MASSIMO UBERTINI

SMALL BASIC

WWW.UBERTINI.IT

MODULO 1

SMALL BASIC

Introduzione
Intellisense
Applicazioni console
Applicazioni GUI
Logo
Musica
Plugin
Conversione in Visual Basic

INTRODUZIONE

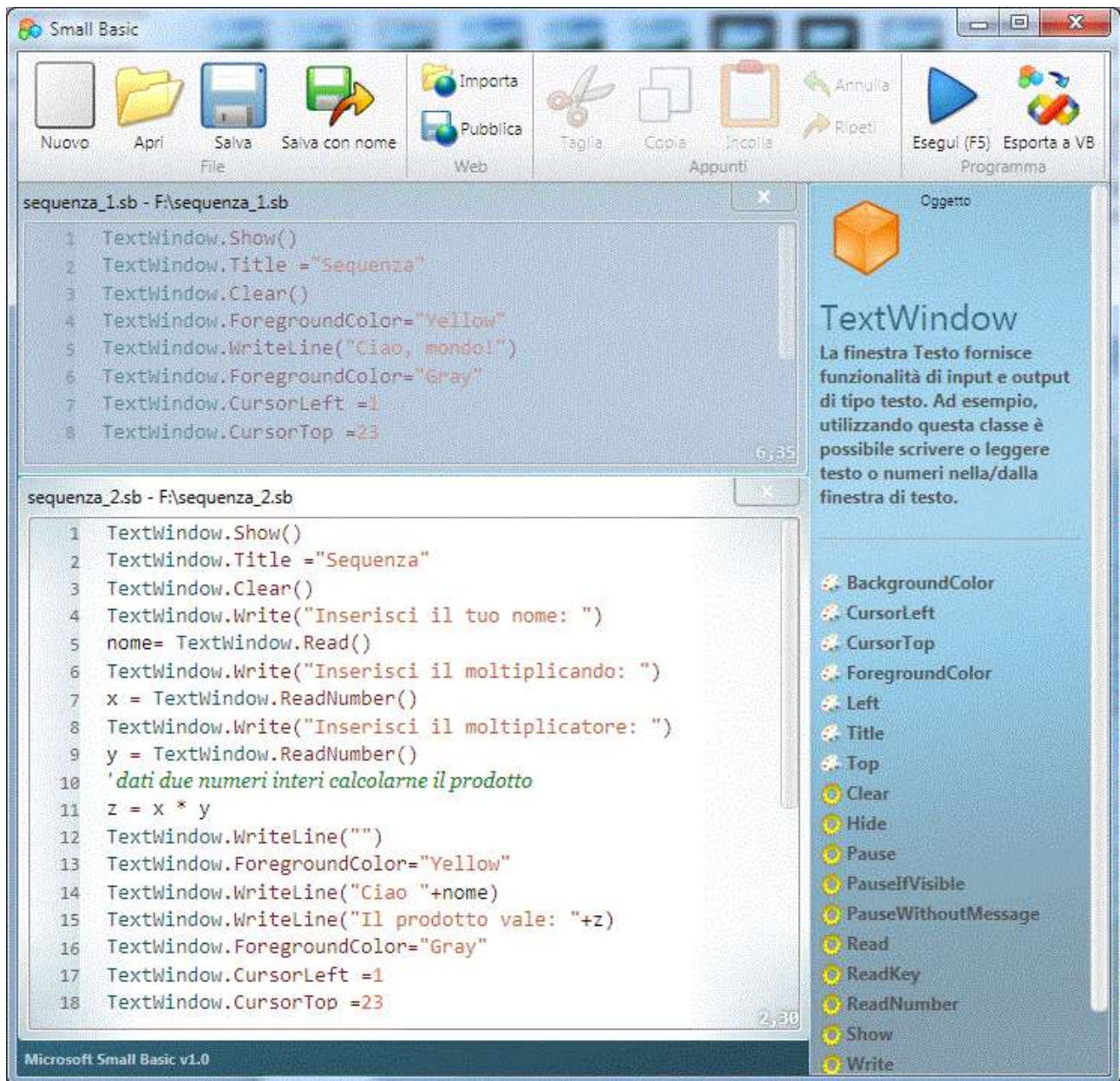
BASIC (*BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE*)

Il linguaggio Basic è stato progettato al Dartmouth College di Hanover Ohio nel 1963 per il calcolatore **GE**-225 (*General Electric*) da John George Kemeny e Thomas Eugene Kurtz.
Il primo programma Basic gira alle 4 del mattino del 01/05/1964.
William Henry Gates III (noto come Bill Gates; Seattle, 28 ottobre 1955) e Paul Gardner Allen (Seattle, 21 gennaio 1953), con l'aiuto di Monte Davidoff, diedero in licenza il loro Basic alla **MITS** (*Micro Instrumentation andTelemetry Systems*) in Albuquerque New Mexico per il computer Altair 8080, la versione occupava un totale di 4 KB di memoria compreso il codice e i dati usati per il codice sorgente, in seguito portarono il loro Basic su altre piattaforme tra cui Apple, Commodore e Atari.
Alla fine degli anni '70 è la volta di **MS-DOS** (*MicroSoft-Disk Operating System*) che è fornito con un interprete Basic.
L'**IBM** (*International Business Machines*) aveva una sua versione denominata **BASICA** (*BASIC Advanced*) mentre la versione fornita con MS-DOS si chiamava **GW-BASIC** (*Graphics and Windows BASIC*) e girava su qualsiasi macchina che potesse far girare anche MS-DOS, ma non c'erano differenze tra i due linguaggi solo che IBM voleva proteggere i suoi PC dai cloni.
Microsoft decise di realizzare anche un compilatore in modo da far girare le applicazioni senza il ricorso all'interprete, la nuova soluzione si chiamò **QBasic** (*QuickBasic*), 18 agosto 1985, che fu distribuito fino alla versione 4.5, poi Microsoft iniziò a distribuire il **PDS Basic** (*Professional Development System*) che terminò con la versione 7.1.
Robert Zale aveva progettato il compilatore Basic che fu poi distribuito dalla Borland Inc. come TurboBasic, ma Zale si mise a distribuire il prodotto anche per conto suo chiamandolo PowerBasic che ora è arrivato alla versione a 32 bit.
La Microsoft, legata per motivi storici e affettivi al Basic, l'ha "riesumato" e l'ha integrato nella tecnologia Object Based realizzando **Visual Basic** nel 1991, che comprende in pratica sia le classiche istruzioni e funzioni del Basic nelle sue numerose varianti sia le istruzioni necessarie per gestire gli oggetti e le proprietà.
Linguaggio di script.

- ✓ **VBA** (*Visual Basic for Applications*) per l'interazione con Microsoft Office.
- ✓ **VBScript** (*Visual Basic Scripting Edition*) è eseguito dal sottosistema **WSH** (*Windows Script Host*) di Windows, anziché dal run-time VB.

Small Basic continua la tradizione della programmazione in linguaggio Basic, è una versione del linguaggio di programmazione Basic sviluppata da Microsoft DevLabs e pubblicata come "Technology Preview" nel 2008.
È pensato per i principianti e ha una struttura molto semplificata rispetto alle altre versioni del Basic, sia come sintassi sia come struttura dell'ambiente di sviluppo didattico, deriva dal QBasic.

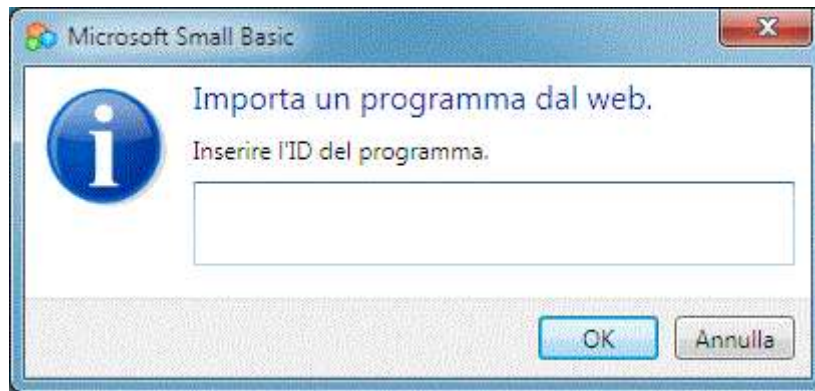
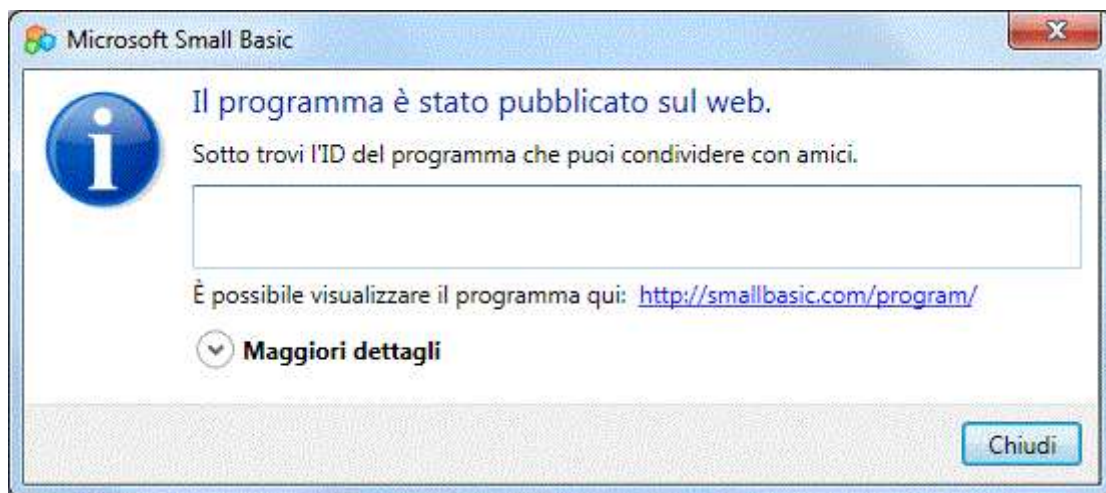
AMBIENTE DI SVILUPPO

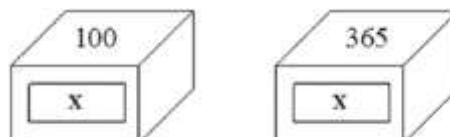
L'ambiente di sviluppo didattico è legato al .NET Framework versione 3.5 **SP** 1 (*Service Pack*) e superiore.
È stato scritto in **WPF** (*Windows Presentation Foundation*).

I pulsanti presenti nella barra degli strumenti, in pratica un controllo ribbon, nella parte superiore della finestra, sono suddivisi in soli quattro gruppi.

1. Gruppo ***File***: comandi ***Nuovo***, ***Apri***, ***Salva*** e ***Salva con nome***.
2. Gruppo ***Web***: comandi ***Importa*** e ***Pubblica***.
3. Gruppo ***Appunti***: pulsanti ***Taglia***, ***Copia***, ***Incolla***, ***Annulla*** e ***Ripeti***.
4. Gruppo ***Programma***: pulsanti ***Esegui (F5)*** ed ***Esporta a VB***.

Nella parte sinistra dell'area principale si trova la finestra dell'editor di codice.

È possibile aprire anche più di una finestra di codice in ogni momento, per lavorare contemporaneamente su più applicazioni o per copiare frammenti di codice da una finestra all'altra.
Nella parte destra della finestra è stata inserita una barra verticale in cui appaiono dinamicamente informazioni sull'operazione che si sta eseguendo in ogni momento.

Nel gruppo ***Web***, fare clic sul pulsante ***Importa***, dopo aver inserito il codice dell'applicazione, è scaricato il codice sorgente direttamente in una finestra dell'editor.

È possibile scrivere un'applicazione ed eseguirla sia sul desktop sia nel browser, grazie al supporto di **Silverlight**, permettendo così di condividere i propri lavori con utilizzatori di tutto il mondo.
Dopo aver progettato un'applicazione, per condividerla fare clic sul pulsante ***Pubblica*** nel gruppo ***Web***, per poterla pubblicare nella galleria ufficiale.

VARIABILE

Rappresenta un valore che varia run-time ed è identificata da un nome, è dinamica.
Una variabile è un contenitore che si trova nella memoria centrale, il valore è il contenuto e il nome è l'etichetta che lo identifica.
Il suo nome, l'etichetta, non può cambiare, ciò che varia è solo il contenuto.
Una variabile di nome X può contenere valori diversi in diversi momenti di tempo.

- ✓ Lunga al massimo 40 caratteri.
- ✓ Include lettere, numeri e underscore.
- ✓ Il primo carattere deve essere una lettera.

InizioTempo, Interesse_Valore, Max07

TIPI DI DATI

Tre tipi di dati.

1. Numerico: intero o decimale.
2. Stringa: insieme di caratteri racchiusi (*" "*).
3. Booleano: due valori *"True"* e *"False"*.

COMMENTI

Iniziano con un apostrofo ('), tutto ciò che segue questo carattere e si trova sulla sua stessa riga è considerato commento.

' questo è un commento

OPERATORI

Aritmetici	**Relazionali**	**Logici**
	Uguale a (=)	
Meno unario (-)	Diverso da (<>)	*And*
Moltiplicazione, Divisione (*, /)	Minore di (<)	*Or*
Divisione intera (\ )	Maggiore di (>)	
	Minore o Uguale a (<=)	
Addizione, Sottrazione (+, -)	Maggiore o Uguale a (>=)	
Stringa concatenata (+).		

Precedenze tra operatori

Nelle espressioni che includono operatori di diverso tipo, i primi ad essere valutati sono gli operatori aritmetici in quest'ordine di precedenza.

Negazione (–)
Moltiplicazione e divisione (*, /)
Divisione tra interi (\)
Addizione e sottrazione (+, –)
Concatenamento di stringhe (&)

Successivamente sono valutati gli operatori relazionali senza una particolare precedenza, ma in ordine da sinistra verso destra e infine gli operatori logici nel seguente ordine.

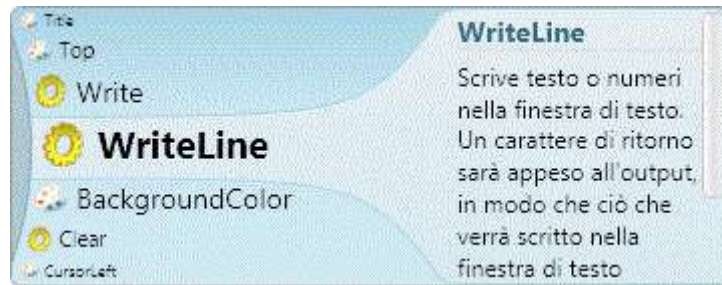
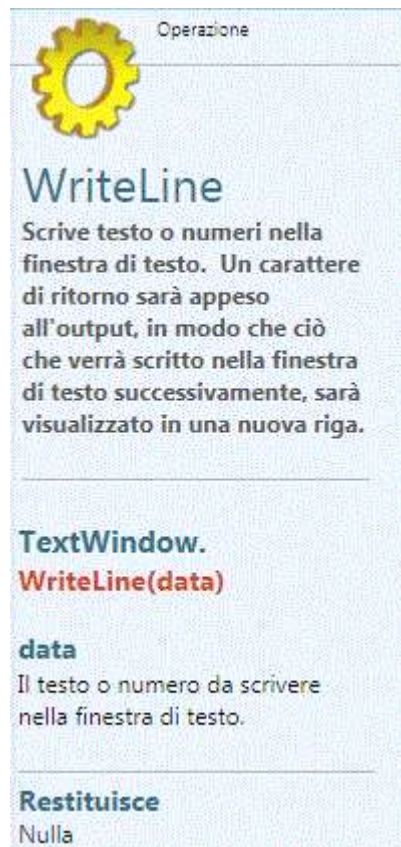
And
Or

AREA DI VISIBILITÀ DELLE VARIABILI

Globale.

INTELLISENSE

INTRODUZIONE

È uno strumento che fornisce informazioni sul contesto attuale: in pratica fornisce ciò che può essere utile, in base a cosa si sta facendo in ogni momento.
È molto intuitivo e comodo, per non dover imparare a memoria nemmeno un'istruzione.
Per esempio, se non si ricorda un elemento del linguaggio o uno degli oggetti disponibili, basta premere la combinazione ***CTRL+Spazio*** e si ottiene un elenco scorrevole.
Con i tasti freccia sarà così possibile scorrere su e giù lungo tutto l'elenco, in modo circolare.

La barra laterale destra fornisce ulteriori informazioni ed esempi di codice, man mano che si scrive nella finestra dell'editor di codice.

Se nell'editor di codice si fa clic su una parola riservata del linguaggio, per esempio sull'**oggetto** *TextWindow*, nella barra laterale destra appaiono tutte le proprietà e tutti i metodi che possono essere utilizzati.
Per esempio, *ForegroundColor* è una **proprietà**, un valore, di *TextWindow*.

In generale, la sintassi è la seguente.

Oggetto.Proprietà = Valore

Per esempio, *Clear* è un **metodo**, un'operazione, di *TextWindow*.
In generale, la sintassi è la seguente.

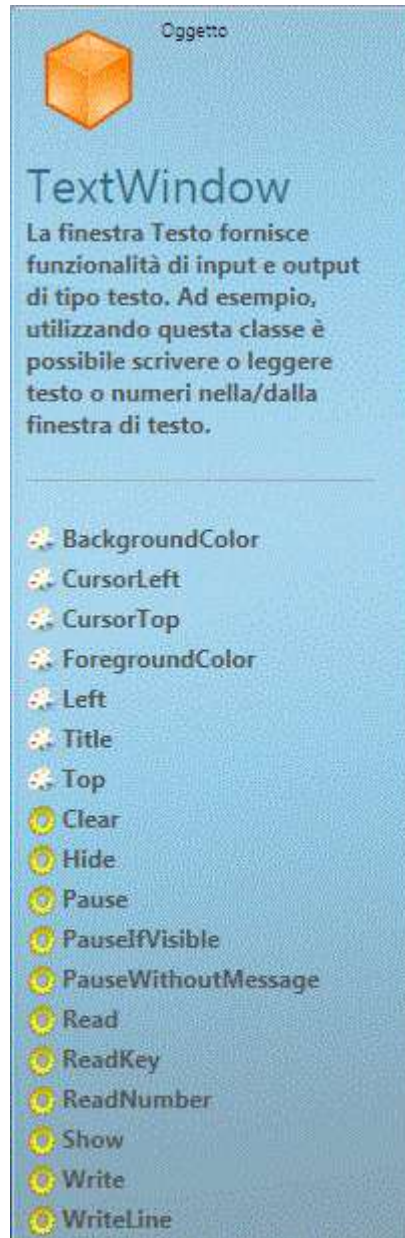
Oggetto.Metodo(Argomenti)
Valore = Oggetto.Metodo(Argomenti)

Gli **eventi** sono come segnali che sono sollevati, per esempio, in risposta alle azioni dell'utente, come muovere il mouse o cliccarlo, sono l'opposto dei metodi.
Nel caso dei metodi, il programmatore li richiama per far fare qualcosa al PC, nel caso degli eventi, il PC fa sapere quando avviene qualcosa d'interessante.
In generale, la sintassi è la seguente.

Oggetto.Evento = EventSub

APPLICAZIONI CONSOLE

INTRODUZIONE

È la finestra della console, una finestra di testo *TextWindow*, simile alla finestra del ***Prompt dei comandi*** di Windows.

Digitare nell'editor il seguente codice.

```
TextWindow.Show()
TextWindow.Title ="Sequenza"
TextWindow.Clear()
TextWindow.ForegroundColor="Yellow"
TextWindow.WriteLine("Ciao, mondo!")
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =0
TextWindow.CursorTop =2
```

Eseguire l’applicazione.

Un’applicazione Small Basic è salvata su disco con estensione **SB** (*Small Basic*).
Questo file non è altro che un file di testo con una diversa estensione: infatti, può essere aperto anche con il ***Blocco note***.
Nel momento in cui si esegue l’applicazione appena salvata, SEQUENZA.SB, sono creati tre file.

1. SEQUENZA.EXE
2. SEQUENZA.PDB
3. SMALLBASICLIBRARY.DLL

PDB (*Program Debug Database*) specifica il nome file delle informazioni di debug.
DLL (*Dynamic Link Library*) libreria a collegamento dinamico.
Questi file, creati dal compilatore, permettono l’esecuzione dell’applicazione senza l’ambiente di sviluppo.

ISTRUZIONI

Un'istruzione è un comando che è eseguito quando l'applicazione è in esecuzione.

Semplici

Non contengono altre istruzioni.

Assegnazione

Una volta che un'espressione è stata calcolata, bisogna memorizzare il suo valore all'interno di una variabile per poterlo utilizzare in seguito.

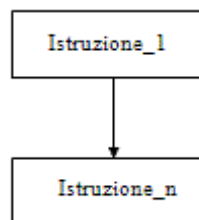
L'operatore (=) assegna ad una variabile posta alla sinistra dell'istruzione, il valore della variabile o dell'espressione posta alla destra.

Variabile = Espressione

Strutturate

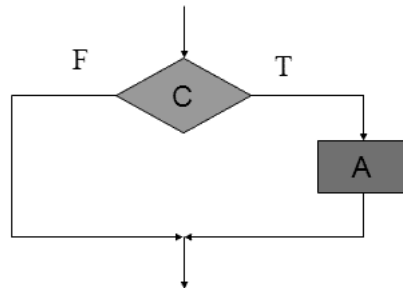
Sono composte da più istruzioni.

Sequenza

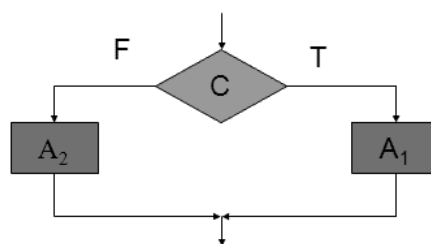
Le istruzioni sono eseguite nello stesso ordine in cui sono scritte.

```
TextWindow.Show()
TextWindow.Title ="Sequenza"
TextWindow.Clear()
TextWindow.Write("Inserisci il tuo nome:       ")
nome= TextWindow.Read()
TextWindow.Write("Inserisci il moltiplicando:  ")
x = TextWindow.ReadNumber()
TextWindow.Write("Inserisci il moltiplicatore: ")
y = TextWindow.ReadNumber()
' dati due numeri interi calcolarne il prodotto
z = x * y
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
TextWindow.WriteLine("Ciao "+nome)
TextWindow.WriteLine("Il prodotto vale: "+z)
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

Selezione
Se si verifica la condizione sceglie una singola o un gruppo d'istruzioni, altrimenti procede in sequenza.

Selezione unaria
Se si verifica la condizione sceglie una singola o un gruppo d'istruzioni.

If** (Condizione) **Then
istruzioni
EndIf

Se è vera la condizione C, **allora** è eseguita l'azione A

Selezione binaria
Se si verifica la condizione sceglie una singola o un gruppo d'istruzioni, altrimenti sceglie l'istruzione o un gruppo dell'altro ramo.

If** (Condizione) **Then
istruzioni1
Else
istruzioni2
EndIf

Se è vera la condizione C, **allora** è eseguita l'azione A_1,
altrimenti è eseguita l'azione A_2

Selezione nidificata
Se la *Condizione1* è vera, sono eseguite le *istruzioni1* tra l'***If*** e il primo ***ElseIf***.
Se la *Condizione1* è falsa, è testata la *Condizione2*, se la *Condizione2* è vera, sono eseguite le *istruzioni2*.
Se la *Condizione2* è falsa, è testata la *Condizione3*.

If** (Condizione1) **Then
istruzioni1
ElseIf** (Condizione2) **Then
istruzioni2

```
ElseIf (Condizione3) Then
        istruzioni3
Else
        istruzioni4
EndIf
```

Esempio, stabilire se un numero è pari o dispari.

```
TextWindow.Show()
TextWindow.Title ="Selezione"
TextWindow.Clear()
TextWindow.Write("Inserisci un numero intero: ")
n = TextWindow.ReadNumber()
r = Math.Remainder (n,2)
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
If (r = 0) Then
        TextWindow.WriteLine("Il numero è pari")
Else
        TextWindow.WriteLine("Il numero è dispari")
EndIf
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

```
Selezione
Inserisci un numero intero: 7
Il numero è dispari
```

Esempio, calcolare il massimo di tre numeri interi.

```
TextWindow.Show()
TextWindow.Title ="Calcolare il massimo di tre numeri interi"
TextWindow.Clear()
TextWindow.Write("Inserisci il primo numero: ")
a = TextWindow.ReadNumber()
TextWindow.Write("Inserisci il primo numero: ")
b = TextWindow.ReadNumber()
TextWindow.Write("Inserisci il primo numero: ")
c = TextWindow.ReadNumber()
max=a
If (max<b) then
        max=b
EndIf
If (max<c) then
        max=c
Endif
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
TextWindow.WriteLine("Il numero maggiore è " + max)
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

Esempio, calcolare il massimo di tre numeri interi utilizzando l'oggetto *Math*.

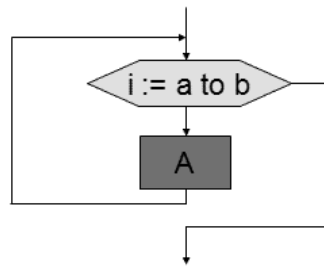
```
TextWindow.Show()
TextWindow.Title ="Calcolare il massimo di tre numeri interi"
TextWindow.Clear()
TextWindow.Write("Inserisci il primo numero: ")
a = TextWindow.ReadNumber()
TextWindow.Write("Inserisci il primo numero: ")
b = TextWindow.ReadNumber()
max= Math.Max (a,b)
TextWindow.Write("Inserisci il primo numero: ")
c = TextWindow.ReadNumber()
max= Math.Max (max,c)
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
TextWindow.WriteLine("Il numero maggiore è " + max)
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

```
Calcolare il massimo di tre numeri interi
Inserisci il primo numero: 3
Inserisci il primo numero: 6
Inserisci il primo numero: 9

Il numero maggiore è 9
```

Iterazione
Successione d’istruzioni eseguite ripetutamente.

Il numero d’iterazioni è noto a priori.

Ciclo enumerativo
Le istruzioni, racchiuse nel ciclo, sono iterate per un numero di volte noto a priori.

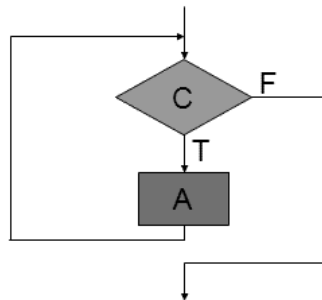
```
For Variabile = Inizio To Fine Step Incremento
        istruzioni
EndFor
```

È un’iterazione di tipo “while”, ma il contatore i avanza automaticamente di 1 ogni ciclo

```
TextWindow.Show()
TextWindow.Title ="Iterazione"
TextWindow.Clear()
TextWindow.ForegroundColor="Yellow"
For i=1 To 10
        TextWindow.WriteLine(i)
EndFor
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

Il numero d'iterazioni non è noto a priori: è il programmatore che deve specificare la condizione che determina la fine dell'iterazione.

Ciclo a condizione iniziale

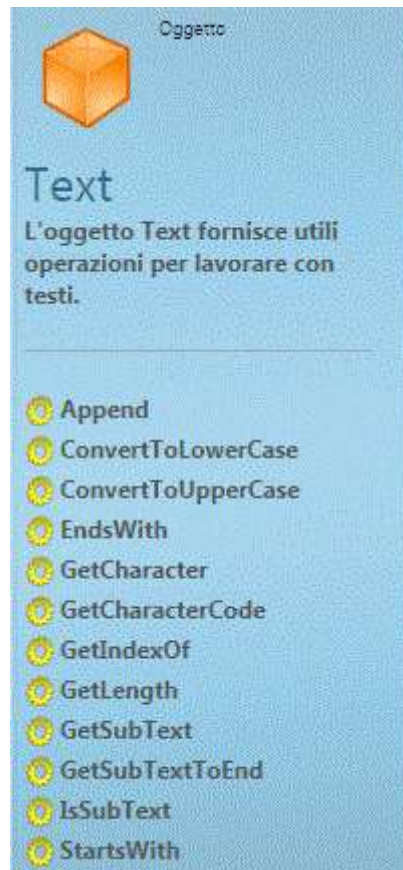
Il blocco è eseguito mentre la condizione, posta all'inizio del ciclo è vera, quando è falsa si esce dal ciclo.

```
While (Condizione)
	istruzioni
EndWhile
```

Mentre è vera la condizione C, **si esegue** l'azione A

```
TextWindow.Show()
TextWindow.Title ="Iterazione"
TextWindow.Clear()
TextWindow.ForegroundColor="Yellow"
i = 1
While (i <= 10)
	TextWindow.WriteLine(i)
	i = i +1
EndWhile
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

Senza fine

```
While ("True")
```

TEXT

Esempio.

```
TextWindow.Show()
TextWindow.Title ="Confronta le stringhe"
TextWindow.Clear()
TextWindow.ForegroundColor="Yellow"
If "ABCD"="abcd" Then
	TextWindow.WriteLine("Stringhe uguali")
Else
	TextWindow.WriteLine("Stringhe diverse")
EndIf
TextWindow.WriteLine(Text.Append ("ABCD","abcd"))
TextWindow.WriteLine(Text.ConvertToLowerCase ("ABCD"))
TextWindow.WriteLine(Text.ConvertToUpperCase ("abcd"))
TextWindow.WriteLine(Text.EndsWith ("ABCD","D"))
TextWindow.WriteLine(Text.GetCharacter(65))
TextWindow.WriteLine(Text.GetCharacterCode("A"))
TextWindow.WriteLine(Text.GetIndexOf ("ABCD","C"))
TextWindow.WriteLine(Text.GetLength ("ABCD"))
TextWindow.WriteLine(Text.GetSubText ("ABCD",2,2))
TextWindow.WriteLine(Text.GetSubTextToEnd ("ABCD",3))
TextWindow.WriteLine(Text.IsSubText ("ABCD","abcd"))
TextWindow.WriteLine(Text.StartsWith ("ABCD","B"))
TextWindow.WriteLine(" ")
TextWindow.ForegroundColor="Yellow"
TextWindow.ForegroundColor="Gray"
```

Confronta le stringhe
Stringhe diverse
ABCDabcd
abcd
ABCD
True
A
65
3
4
BC
CD
False
False

ARRAY

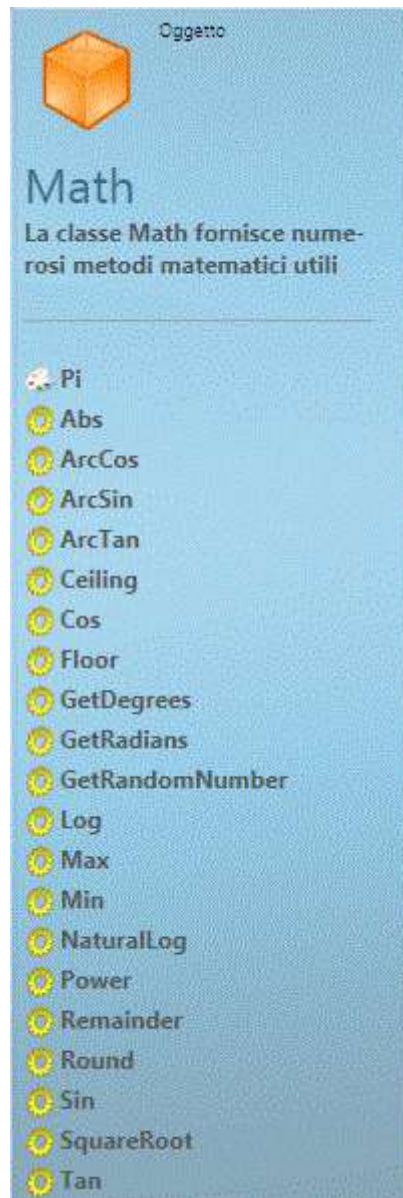
Esempio.

```
TextWindow.Show()
TextWindow.Title ="Array"
TextWindow.Clear()
TextWindow.Write("Numero di elementi dell'array: ")
n = TextWindow.ReadNumber()
TextWindow.WriteLine("")
' caricamento array
For i = 1 To n
	TextWindow.Write("Elemento Numero N° " + i + ", inserisci un numero intero: ")
	vet[i] = TextWindow.Read ()
EndFor
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
' stampa array
For i = 1 To n
	TextWindow.WriteLine(vet[i])
EndFor
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

```
Array
Numero di elementi da caricare [<=10]: 4
Numero N° 1, un numero intero: 55
Numero N° 2, un numero intero: 66
Numero N° 3, un numero intero: 77
Numero N° 4, un numero intero: 88

55
66
77
88
```

Esempio, matrice.

```
TextWindow.Show()
TextWindow.Title ="Matrice"
TextWindow.Clear()
TextWindow.Write("Numero di righe della matrice:   ")
righe = TextWindow.ReadNumber()
TextWindow.Write("Numero di colonne della matrice: ")
colonne = TextWindow.ReadNumber()
TextWindow.WriteLine("")
' caricamento matrice
For riga = 1 To righe
   For colonna = 1 To colonne
        TextWindow.Write("Elemento N° " + riga+colonna+ " inserisci un numero intero: ")
        mat[riga][colonna] = TextWindow.Read ()
   EndFor
EndFor
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
' stampa matrice
For riga = 1 To righe
        For colonna = 1 To colonne
                TextWindow.Write(mat[riga][colonna]+" ")
        EndFor
        TextWindow.WriteLine("")
EndFor
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

```
Matrice
Numero di righe della matrice:   3
Numero di colonne della matrice: 3

Elemento N° 11 inserisci un numero intero: 1
Elemento N° 12 inserisci un numero intero: 2
Elemento N° 13 inserisci un numero intero: 3
Elemento N° 21 inserisci un numero intero: 4
Elemento N° 22 inserisci un numero intero: 5
Elemento N° 23 inserisci un numero intero: 6
Elemento N° 31 inserisci un numero intero: 7
Elemento N° 32 inserisci un numero intero: 8
Elemento N° 33 inserisci un numero intero: 9

1 2 3
4 5 6
7 8 9
```

Subroutine

È una porzione di codice all'interno di un'applicazione che fa qualcosa di specifico e che può essere chiamata in qualunque punto dell'applicazione.

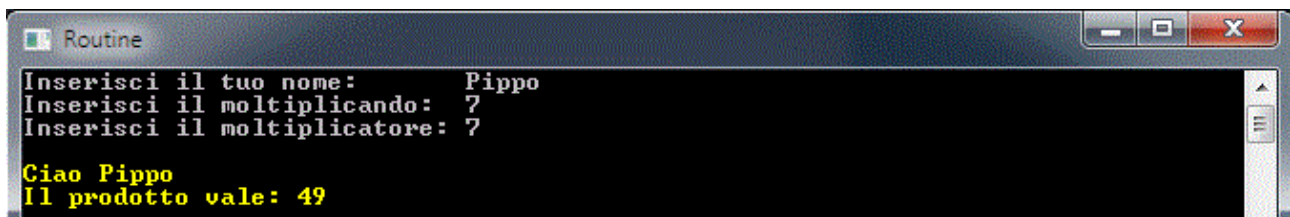
Vantaggi delle routine.

- ✓ Riducono la quantità di codice che occorre digitare.
- ✓ Riducono la complessità dei problemi.
- ✓ Migliorano la leggibilità di un'applicazione.

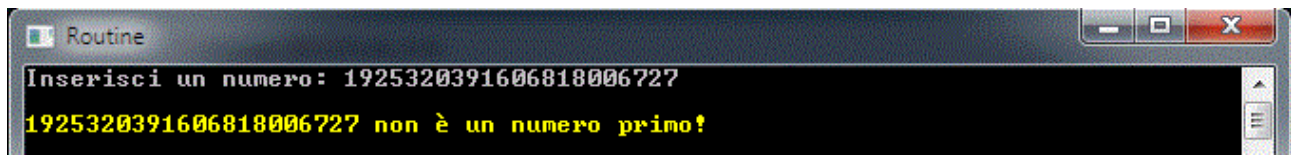
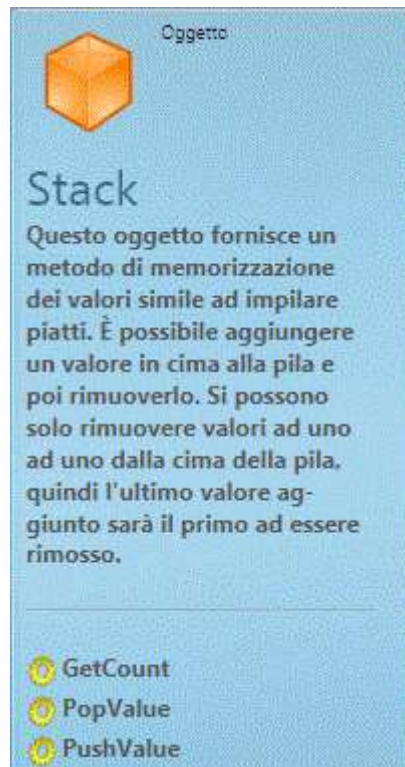
```
Sub MiaSubroutine
        istruzioni
EndSub
```

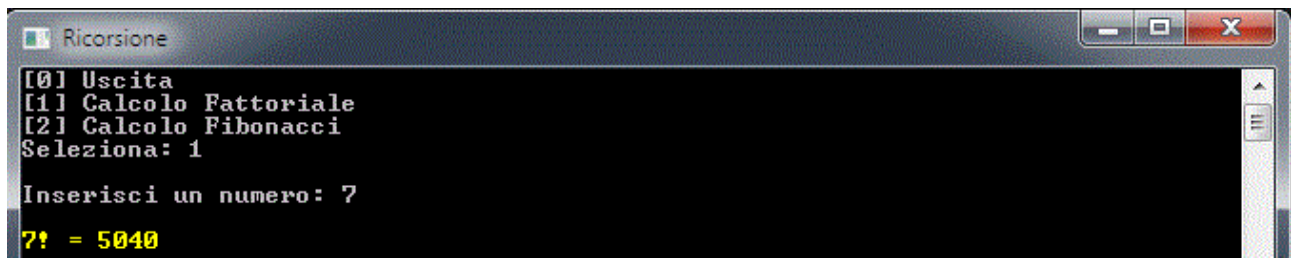
```
TextWindow.Show()
TextWindow.Title ="Routine"
TextWindow.Clear()
TextWindow.Write("Inserisci il tuo nome:       ")
nome= TextWindow.Read()
TextWindow.Write("Inserisci il moltiplicando:  ")
```

```
x = TextWindow.ReadNumber()
TextWindow.Write("Inserisci il moltiplicatore: ")
y = TextWindow.ReadNumber()
Prodotto()
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
TextWindow.WriteLine("Ciao "+nome)
TextWindow.WriteLine("Il prodotto vale: "+z)
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
Sub Prodotto
	' dati due numeri interi calcolarne il prodotto
	z = x * y
EndSub
```

```
Routine
Inserisci il tuo nome:        Pippo
Inserisci il moltiplicando:  7
Inserisci il moltiplicatore: 7

Ciao Pippo
Il prodotto vale: 49
```

Esempio, progettare un'applicazione che dato in input un numero intero, esegua il test di primalità.

```
TextWindow.Show()
TextWindow.Title ="Routine"
TextWindow.Clear()
TextWindow.Write("Inserisci un numero: ")
i = TextWindow.ReadNumber()
TextWindow.ForegroundColor="Yellow"
isPrime = "True"
Primi()
TextWindow.WriteLine("")
If (isPrime = "True") Then
	TextWindow.WriteLine(i + " è un numero primo!")
Else
	TextWindow.WriteLine(i + " non è un numero primo!")
Endif
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
Sub Primi
	For j = 2 To Math.SquareRoot (i)
		If (Math.Remainder (i, j) = 0) Then
			isPrime = "False"
			Goto fine
		EndIf
	EndFor
	fine:
EndSub
```

Routine
Inserisci un numero: 1925320391606818006727
1925320391606818006727 non è un numero primo!

STACK

Esempio, ricorsione.

```
Sub Main
 n   = 1
 sel = 1
 TextWindow.Show()
 TextWindow.Title ="Ricorsione"
 While(sel<>0)
  TextWindow.Clear()
  TextWindow.WriteLine("[0] Uscita")
  TextWindow.WriteLine("[1] Calcolo Fattoriale")
  TextWindow.WriteLine("[2] Calcolo Fibonacci")
  TextWindow.Write("Seleziona: ")
  sel=TextWindow.ReadNumber()
  TextWindow.WriteLine("")
  If(sel>0 And sel<3)Then
   TextWindow.Write("Inserisci un numero: ")
   n=TextWindow.ReadNumber()
  EndIf
  Stack.PushValue("Heap",n)   ' salvo la variabili locale n
  Stack.PushValue("Heap",n)   ' sia per Fact() sia per Fib()
  If(sel=1)Then               ' calcolo il fattoriale
   Fact()
  ElseIf(sel=2)Then           ' calcolo Fibonacci
   Fib()
  EndIf
  ret = Stack.PopValue("Heap")' valore di ritorno
  n   = Stack.PopValue("Heap")' restore della variabile locale
  TextWindow.WriteLine("")
  TextWindow.ForegroundColor="Yellow"
```

```
  If(sel=0)Then
   ElseIf(sel=1)Then          ' risultato del fattoriale
     TextWindow.WriteLine(n + "! = " + ret)
     TextWindow.Readkey()
  ElseIf(sel=2)Then           ' risultato Fibonacci
    TextWindow.WriteLine("Fib(" + n + ") = " + ret)
    TextWindow.Readkey()
  Else
    TextWindow.WriteLine("Scelta errata!")
    TextWindow.Readkey()
  EndIf
  TextWindow.ForegroundColor="Gray"
  TextWindow.CursorLeft =1
  TextWindow.CursorTop =23
 EndWhile
EndSub
' calcolo del fattoriale
Sub Fact
 n=Stack.PopValue("Heap")      ' legge il parametro
 If(n<0)Then
  TextWindow.WriteLine("Errore: il numero è negativo!")
  Stack.PushValue("Heap",0)
 ElseIf(n=0)Then               ' il fattoriale di 0 è 1
  Stack.PushValue("Heap",1)   ' valore di ritorno
 Else
  Stack.PushValue("Heap",n)   ' salvo la variabile locale
  Stack.PushValue("Heap",n-1) '  parametro per la chiamata del fattoriale
  Fact()                     ' ricorsione per (n-1)!
                            ' valore di ritorno con n * (n-1)!
  Stack.PushValue("Heap",Stack.PopValue("Heap")*Stack.PopValue("Heap"))
 EndIf
EndSub
' calcolo Fibonacci
Sub Fib
 n = Stack.PopValue("Heap")   ' legge il parametro
 If(n=0)Then
  Stack.PushValue("Heap",0)  ' valore di ritorno per n=0
 ElseIf(n=1)Then
  Stack.PushValue("Heap",1)  ' valore di ritorno per n=1
 Else
  Stack.PushValue("Heap",n)  ' salvo la variabile locale
  Stack.PushValue("Heap",n-1)' parametro per la chiamata Fib()
  Fib()                     ' ricorsione  Fib(n-1)
  ret=Stack.PopValue("Heap") ' risultato
  n=Stack.PopValue("Heap")   ' Restore local variable n
  Stack.PushValue("Heap",ret)' salvo la variabile locale Ret, n non è più necessario
  Stack.PushValue("Heap",n-2)' parametro per Fib()
  Fib()                     ' ricorsione Fib(n-2)
                           ' valore di ritorno Fib(n-1)+Fib(n-2)
  Stack.PushValue("Heap",Stack.PopValue("Heap")+Stack.PopValue("Heap"))
 EndIf
EndSub
TextWindow.Show()
TextWindow.Title ="Ricorsione"
```

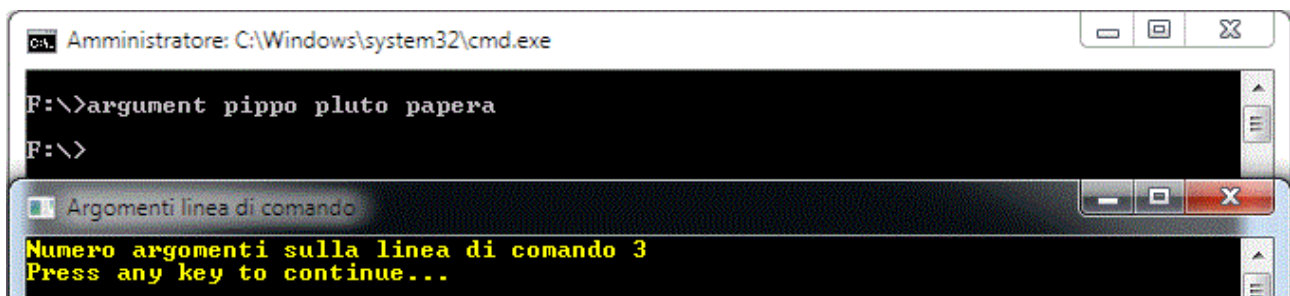
Main()

Ricorsione

```
[0] Uscita
[1] Calcolo Fattoriale
[2] Calcolo Fibonacci
Seleziona: 1

Inserisci un numero: 7

7! = 5040
```

PROGRAM

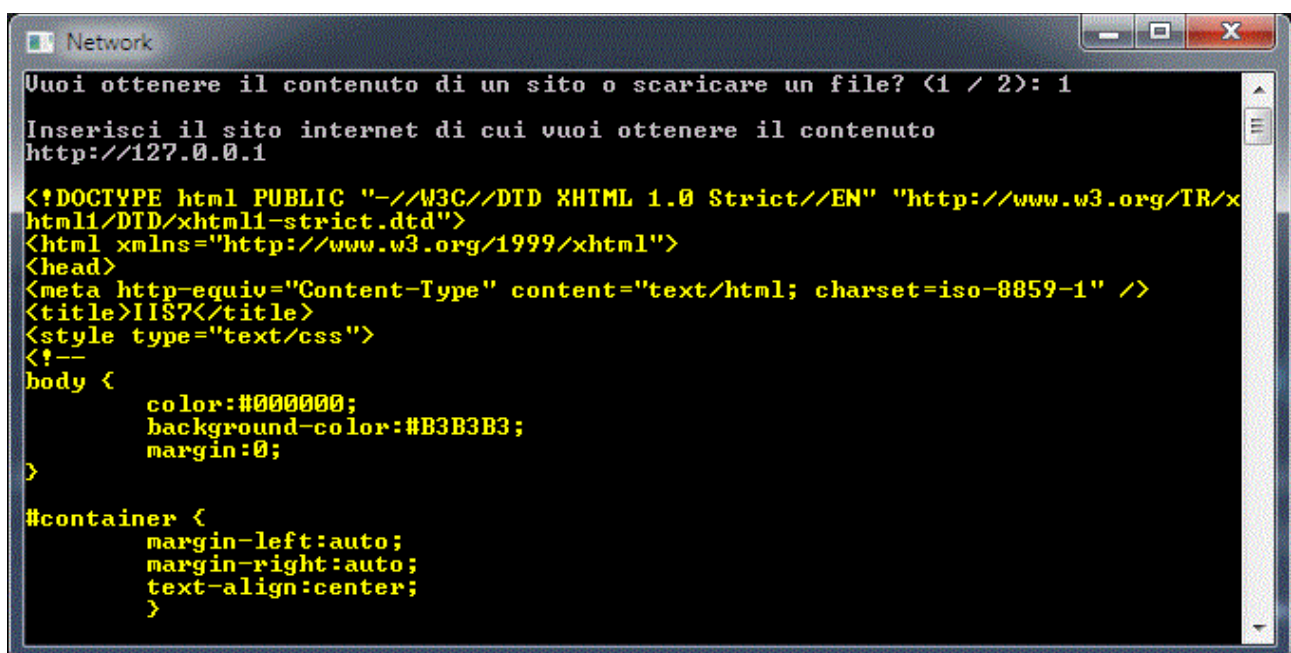
Esempio.

```
TextWindow.Show()
TextWindow.Title ="Argomenti linea di comando"
TextWindow.Clear()
TextWindow.ForegroundColor="Yellow"
TextWindow.WriteLine("Numero argomenti sulla linea di comando "  +
Program.ArgumentCount)
```

NETWORK

Esempio.

```
TextWindow.Show()
TextWindow.Title ="Network"
TextWindow.Clear()
TextWindow.ForegroundColor="Yellow"
TextWindow.Write("Vuoi ottenere il contenuto di un sito o scaricare un file? (1 / 2): ")
risp = TextWindow.Read()
If(risp = "1") then
  TextWindow.WriteLine("Inserisci il sito internet di cui vuoi ottenere il contenuto")
  url = TextWindow.Read()
  Pagina =Network.GetWebPageContents(url)
  TextWindow.WriteLine(Pagina)
EndIf
If(risp = "2") then
  TextWindow.WriteLine("Inserisci il link di download")
  link = TextWindow.Read()
  download = Network.DownloadFile(link)
EndIf
```

```
Network
Vuoi ottenere il contenuto di un sito o scaricare un file? (1 / 2): 1

Inserisci il sito internet di cui vuoi ottenere il contenuto
http://127.0.0.1

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/x
html1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS7</title>
<style type="text/css">
<!--
body {
        color:#000000;
        background-color:#B3B3B3;
        margin:0;
}

#container {
        margin-left:auto;
        margin-right:auto;
        text-align:center;
        }
```

DICTIONARY

Esempio.

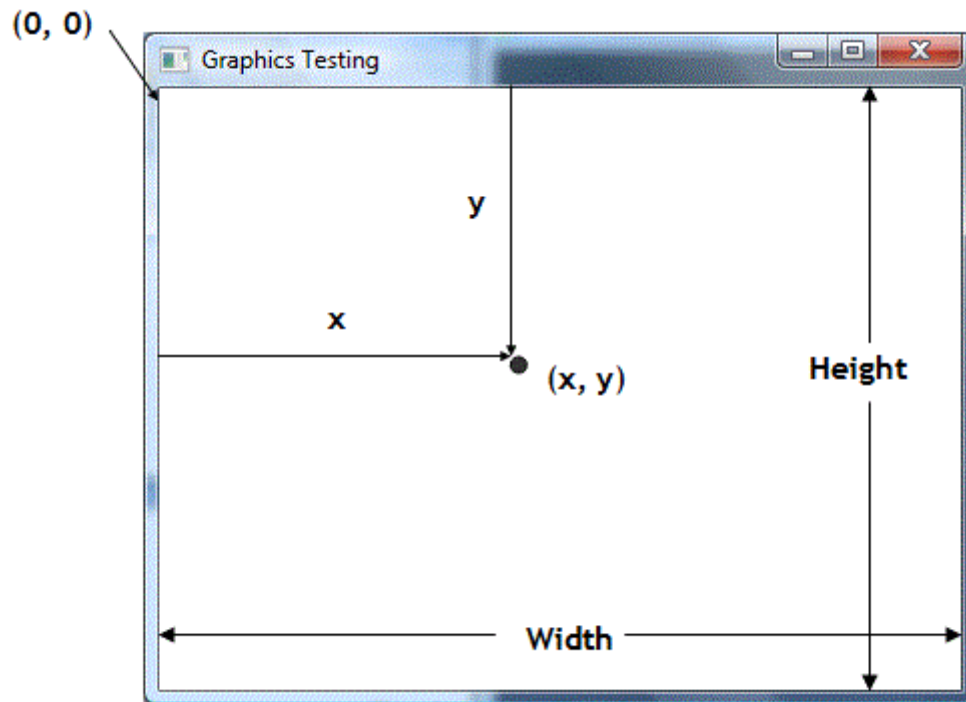
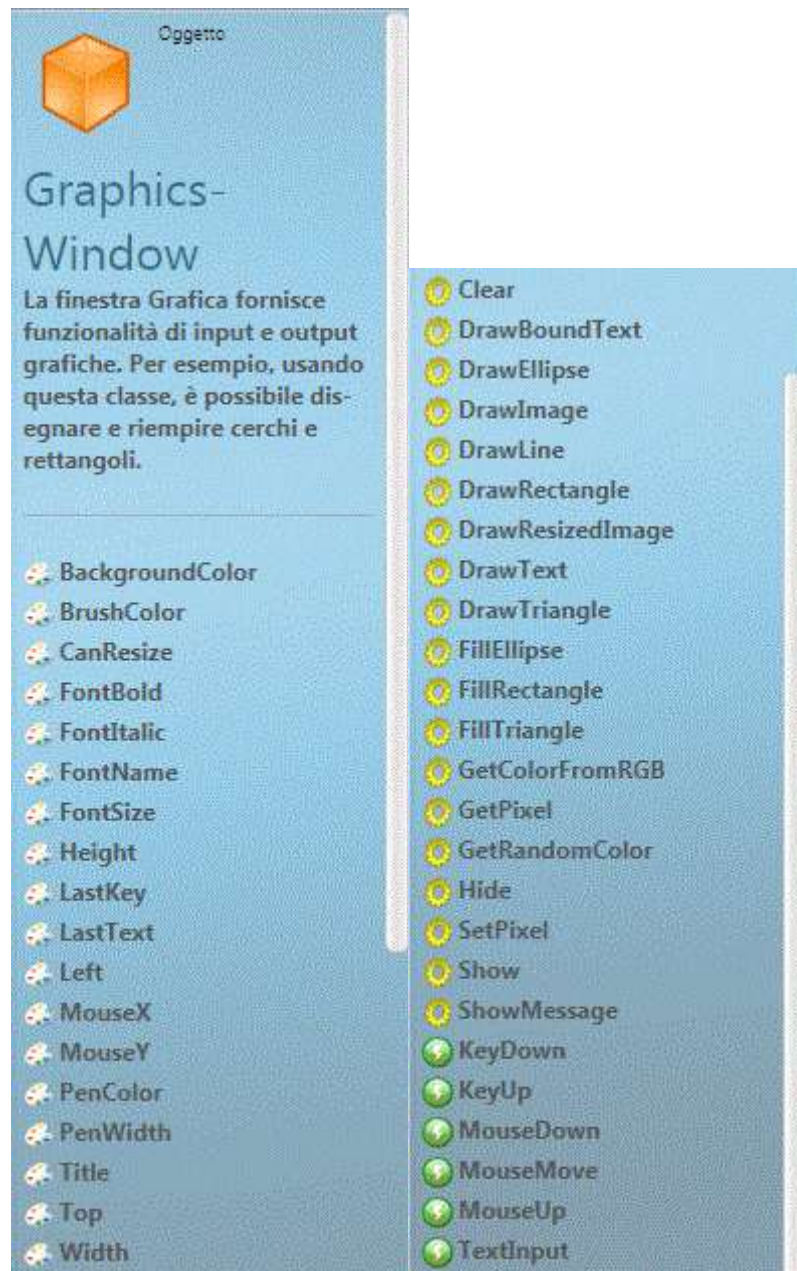
```
TextWindow.Show()
TextWindow.Title = ("Dizionario")
TextWindow.Clear()
TextWindow.ForegroundColor="Yellow"
TextWindow.Write("Digita una parola: ")
word = TextWindow.Read()
TextWindow.Hide()
GraphicsWindow.Title = ("Definizione  "+word+"")
GraphicsWindow.FontBold = "False"
GraphicsWindow.FontName = "Verdana"
GraphicsWindow.FontSize = 24
GraphicsWindow.BrushColor = "DarkGreen"
GraphicsWindow.DrawText(10, 10, word)
definition = Dictionary.GetDefinition(word)
GraphicsWindow.FontSize = 12
GraphicsWindow.BrushColor = "Black"
GraphicsWindow.DrawText(10, 60, definition)
```

Esempio.

```
Variabile = 1
TextWindow.ForegroundColor= "Green"
TextWindow.Title = ("Dizionario SmallBasic")
While Variabile = 1
  TextWindow.WriteLine("Inserisci parola di cui vuoi conoscere la definizione:")
  TextWindow.WriteLine("")
Parola = TextWindow.Read()                      ' parola inserita
definition = Dictionary.GetDefinition(Parola)   ' funzione dizionario
TextWindow.WriteLine(definition)                ' scrive definizione
If(definition = "") Then                        ' caso in cui non è trovata la definizione
  TextWindow.WriteLine("Non è stata trovata nessuna definizione!")
  TextWindow.WriteLine("")
  EndIf
EndWhile
```

APPLICAZIONI GUI

INTRODUZIONE

Il sistema di coordinate identifica un singolo pixel (x, y) della finestra grafica.
La coordinata orizzontale (x) s'incrementa, a partire da 0, da sinistra verso destra.
La coordinata verticale (y) s'incrementa, a partire da 0, dal basso verso l'alto.

È la finestra grafica, *GraphicsWindow*, **GUI** (*Graphics User Interface*).

MOUSE

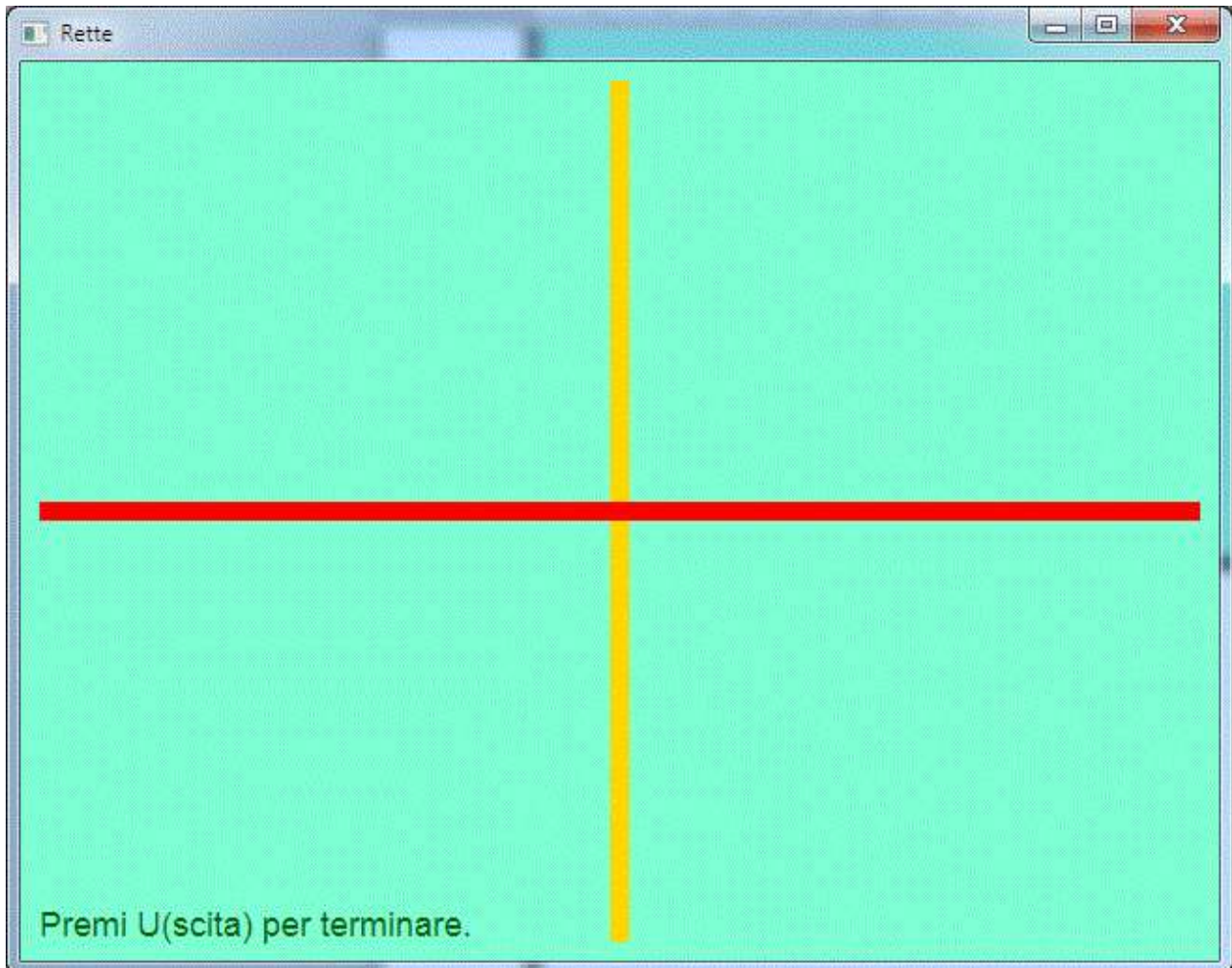
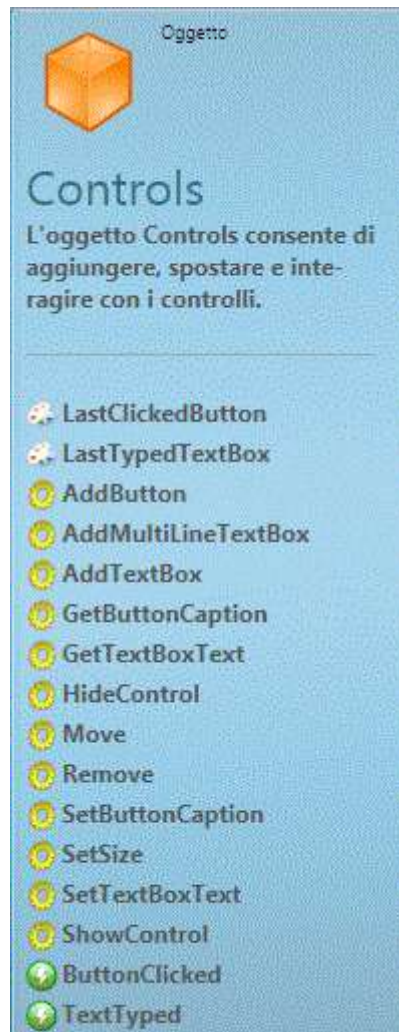
Esempio, prima finestra GUI.
La prima linea assegna il nome della routine all'evento *KeyDown* dell'oggetto *Window*. *KeyDown* assomiglia ad una proprietà, però invece di assegnare un valore, si assegna la routine *OnKeyDown*; quando si verifica l'evento, la routine è chiamata automaticamente. Nell'esempio, la routine *OnKeyDown* è chiamata ogni volta che l'utente digita sulla tastiera la lettera *"U"* per terminare l'applicazione.
Le applicazioni scritte in questo modo sono chiamate **applicazioni event-driven**.

```
GraphicsWindow.KeyDown = OnKeyDown
presenta()
stampa()
Sub presenta
	GraphicsWindow.BackgroundColor = "Aquamarine"
	GraphicsWindow.Title = "Prima applicazione"
	GraphicsWindow.Width = 640
	GraphicsWindow.Height = 480
	GraphicsWindow.Show()
	GraphicsWindow.Clear()
EndSub
Sub stampa
	GraphicsWindow.FontBold = "False"
	GraphicsWindow.FontName = "Arial"
	GraphicsWindow.FontSize = 18
	GraphicsWindow.BrushColor = "DarkGreen"
	GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub OnKeyDown
	If (GraphicsWindow.LastKey = "U") Then
		Program.End()
	EndIf
EndSub
```

Esempio, disegnare rette.

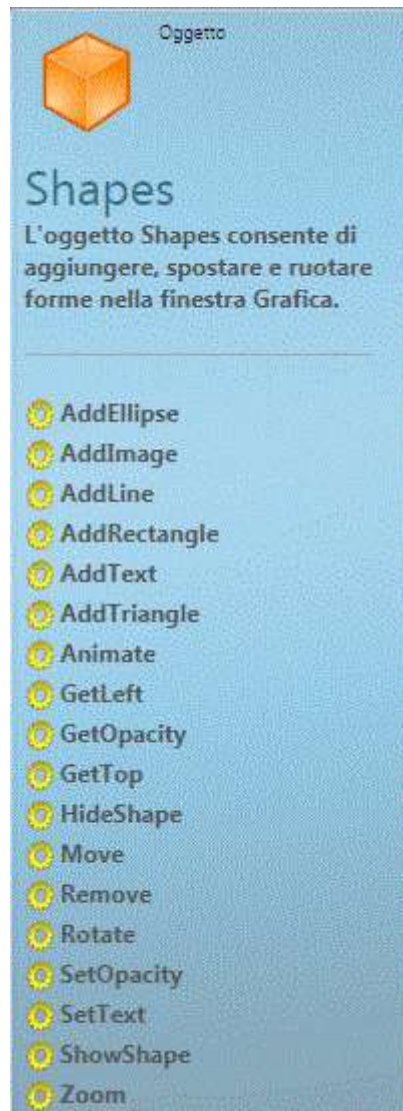
```
GraphicsWindow.KeyDown = OnKeyDown
presenta()
calcola()
stampa()
Sub calcola
	GraphicsWindow.PenWidth = 10
	GraphicsWindow.PenColor ="Gold"
	GraphicsWindow.DrawLine (320,10,320,470)
	GraphicsWindow.PenColor ="Red"
	GraphicsWindow.DrawLine (10,240,630,240)
EndSub
Sub presenta
	GraphicsWindow.BackgroundColor = "Aquamarine"
	GraphicsWindow.Title = "Rette"
	GraphicsWindow.Width = 640
	GraphicsWindow.Height = 480
	GraphicsWindow.Show()
	GraphicsWindow.Clear()
EndSub
Sub stampa
	GraphicsWindow.FontBold = "False"
	GraphicsWindow.FontName = "Arial"
```

```
        GraphicsWindow.FontSize = 18
        GraphicsWindow.BrushColor = "DarkGreen"
        GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub OnKeyDown
        If (GraphicsWindow.LastKey = "U") Then
                Program.End()
        EndIf
EndSub
```

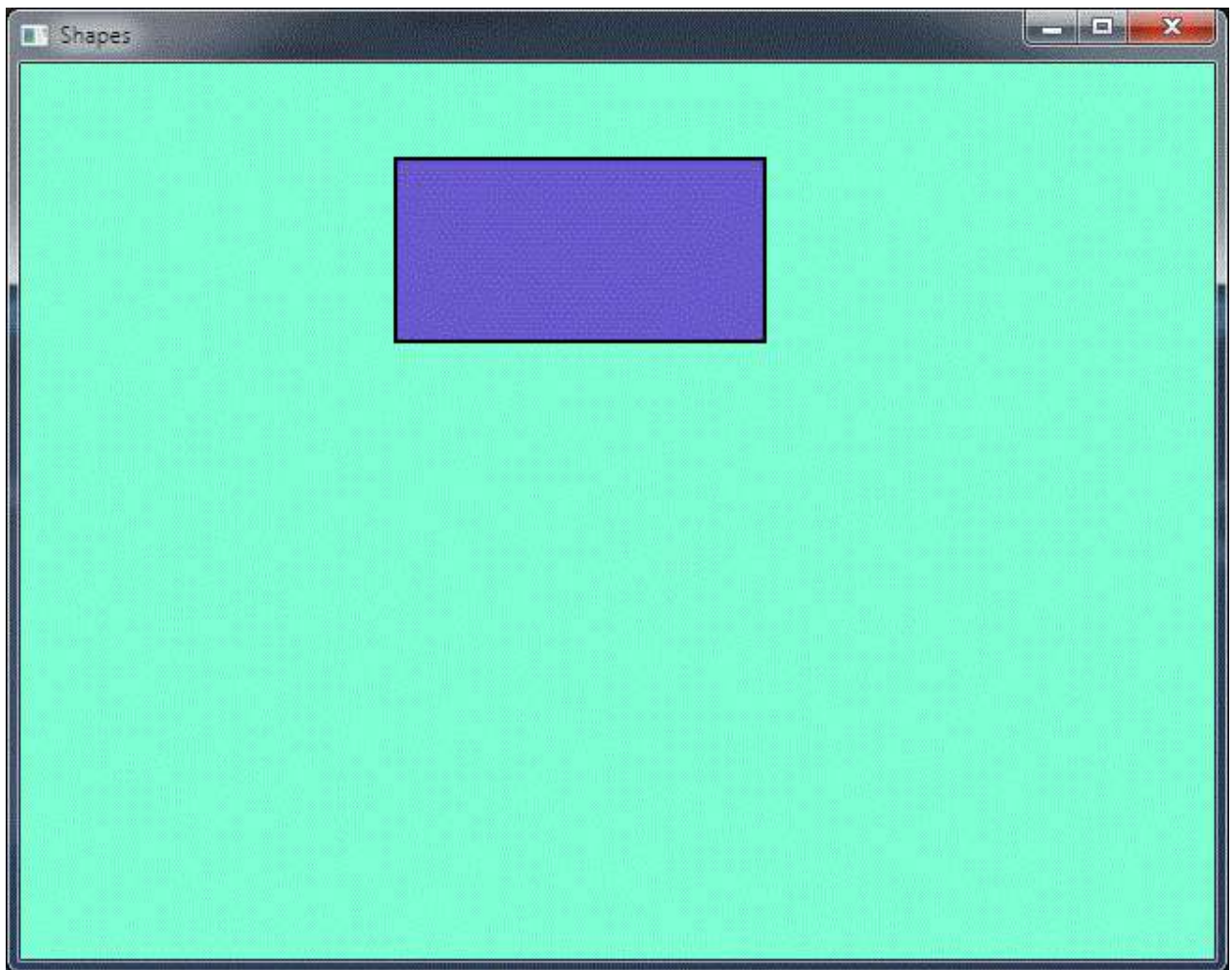
CONTROLS

Esempio, moltiplicare due numeri interi.

```
presenta()
begin:
calcola()
stampa()
Goto begin
Sub presenta
        GraphicsWindow.BackgroundColor = "Aquamarine"
        GraphicsWindow.Title = "Prodotto"
        GraphicsWindow.Width = 640
        GraphicsWindow.Height = 480
        GraphicsWindow.Show()
        GraphicsWindow.Clear()
        GraphicsWindow.BrushColor = "Black"
        GraphicsWindow.DrawText(10,10,"Inserisci il moltiplicando: ")
        txtPrimo =Controls.AddTextBox (200,10)
        GraphicsWindow.DrawText(10,40,"Inserisci il moltiplicatore: ")
        txtSecondo =Controls.AddTextBox (200,40)
        btnCalc = Controls.AddButton("Il prodotto vale: ",10,80)
        txtRisultato =Controls.AddTextBox (200,80)
EndSub
```

```
Sub calcola
	x = Controls.GetTextBoxText(txtPrimo)
	y = Controls.GetTextBoxText(txtSecondo)
	z = x * y
EndSub
Sub stampa
	GraphicsWindow.FontBold = "False"
	GraphicsWindow.FontName = "Arial"
	GraphicsWindow.FontSize = 18
	GraphicsWindow.BrushColor = "Red"
	If Controls.LastClickedButton = btnCalc then
		Controls.SetTextBoxText (txtRisultato,z)
	endif
	GraphicsWindow.BrushColor = "DarkGreen"
	GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub OnKeyDown
	If (GraphicsWindow.LastKey = "U") Then
		Program.End()
	endif
EndSub
```

SHAPES

Esempio.

```
GraphicsWindow.BackgroundColor = "Aquamarine"
GraphicsWindow.Title = "Shapes"
GraphicsWindow.Width = 640
GraphicsWindow.Height = 480
GraphicsWindow.Show()
rectangle = Shapes.AddRectangle(200, 100)
Shapes.Move(rectangle, 100, 50)
For i = 1 To 10
  Shapes.ShowShape(rectangle)
  Shapes.Move(rectangle, 50, 50)
  Shapes.Animate(rectangle, 20 * I, 50, 5000)
EndFor
```

Shapes

DESKTOP

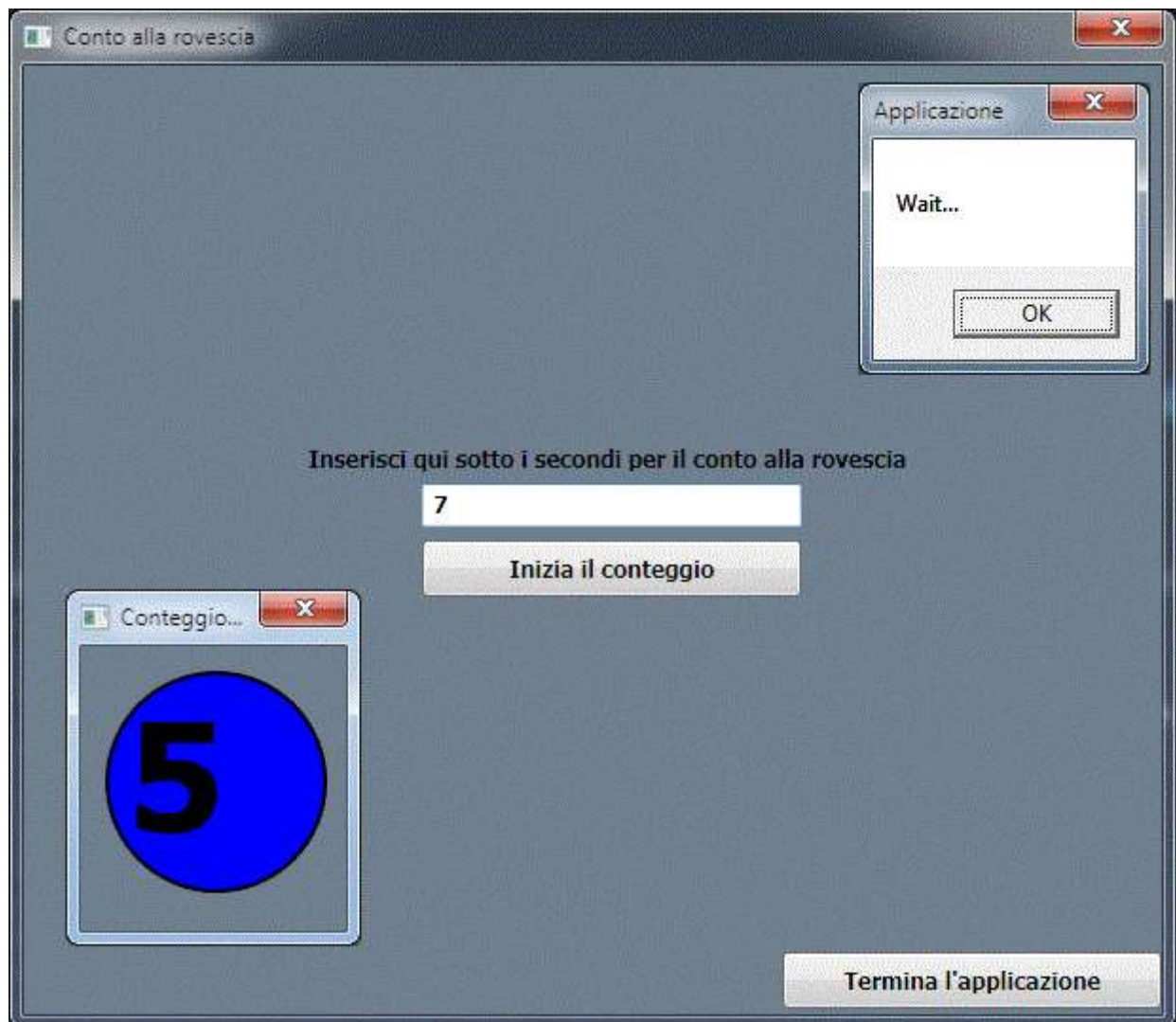
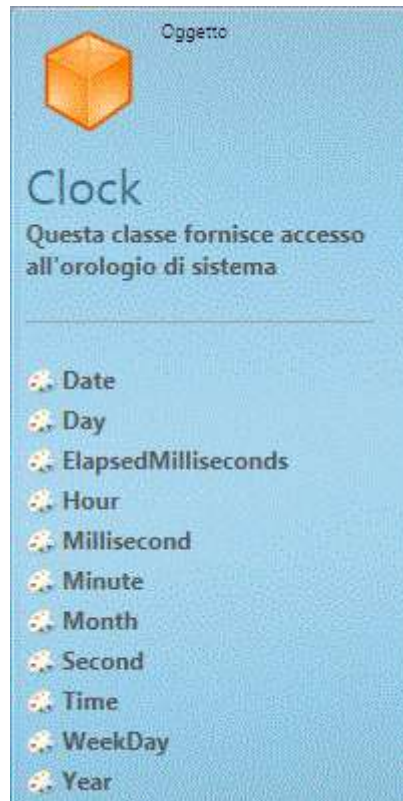
Esempio.

```
' Flickr è un sito web multilingua che permette agli iscritti di condividere
' fotografie personali con chiunque abbia accesso a Internet,
' in un ambiente web 2.0, fondato nel 2002, oggi Flickr conta diverse
' migliaia di fotografie caricate ed è in continua crescita
While ("True")
  foto = Flickr.GetRandomPicture("bike")      ' tag di ricerca
  Desktop.SetWallPaper(foto)                  ' imposta l'immagine come desktop
  Program.Delay(3 * 1000)                     ' tempo(secondi) per cambiare immagine
EndWhile
```

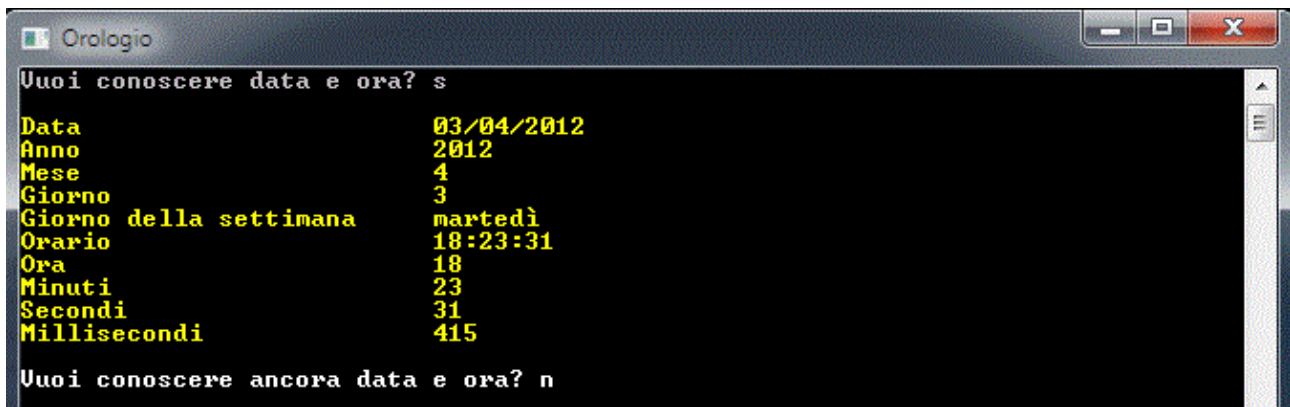
TIMER

Esempio, progettare un conto alla rovescia.

```
GraphicsWindow.BackgroundColor = "SlateGray"
GraphicsWindow.BrushColor = "Black"
GraphicsWindow.CanResize ="False"
GraphicsWindow.Top  =  100
GraphicsWindow.Left =  300
controlli()
begin:
 If( Controls.LastClickedButton = cont)Then
   conteggio()
   EndIf
   If(Controls.LastClickedButton=end)Then
   GraphicsWindow.ShowMessage("Wait...","Applicazione")
   Program.Delay(1500)
   Program.End()
 EndIf
 Goto begin
Sub controlli
 GraphicsWindow.Clear()
 GraphicsWindow.FontSize = 12
 GraphicsWindow.Width = 600
 GraphicsWindow.Height= 500
 GraphicsWindow.Title="Conto alla rovescia"
 GraphicsWindow.DrawBoundText(150,200,335,"Inserisci qui sotto i secondi per il conto
alla rovescia")
 input = Controls.AddTextBox(210 , 220)
 Controls.SetSize(input,200,23)
 cont=Controls.AddButton("Inizia il conteggio",210,250)
 Controls.SetSize(cont,200,30)
 end =Controls.AddButton("Termina l'applicazione",400,467)
 Controls.SetSize(end,200,30)
EndSub
Sub conteggio
```

```
 sec = Controls.GetTextBoxText(input)
 GraphicsWindow.Title="Conteggio..."
 GraphicsWindow.Width = 140
 GraphicsWindow.Height= 150
 For c = 0 To sec
   GraphicsWindow.Clear()
   GraphicsWindow.BrushColor = "blue"
   GraphicsWindow.FillEllipse( 14 , 14 , 115 , 115)
   GraphicsWindow.DrawEllipse( 14 , 14 , 115 , 115)
   GraphicsWindow.BrushColor = "Black"
   GraphicsWindow.FontSize = 80
   GraphicsWindow.DrawText(25,18,sec-c)
   Program.Delay(1000)
   If(sec=c)Then
     Sound.PlayBellRing()
     controlli()
     Goto begin
   EndIf
 EndFor
EndSub
```

CLOCK

Esempio.

```
TextWindow.Show()
TextWindow.Title ="Orologio"
TextWindow.Clear()
TextWindow.Write("Vuoi conoscere data e ora? ")
risp = TextWindow.Read()
TextWindow.ForegroundColor="Yellow"
While(risp = "s")
  TextWindow.WriteLine("")
  TextWindow.WriteLine("Data                    " + Clock.Date)
  TextWindow.WriteLine("Anno                    " + Clock.Year)
  TextWindow.WriteLine("Mese                    " + Clock.Month)
  TextWindow.WriteLine("Giorno                  " + Clock.Day)
  TextWindow.WriteLine("Giorno della settimana  " + Clock.WeekDay)
  TextWindow.WriteLine("Orario                  " + Clock.Time)
  TextWindow.WriteLine("Ora                     " + Clock.Hour)
  TextWindow.WriteLine("Minuti                  " + Clock.Minute)
  TextWindow.WriteLine("Secondi                 " + Clock.Second)
  TextWindow.WriteLine("Millisecondi            " + Clock.Millisecond)
  TextWindow.WriteLine("")
  TextWindow.ForegroundColor="White"
  TextWindow.Write("Vuoi conoscere ancora data e ora? ")
  risp = TextWindow.Read()
EndWhile
```

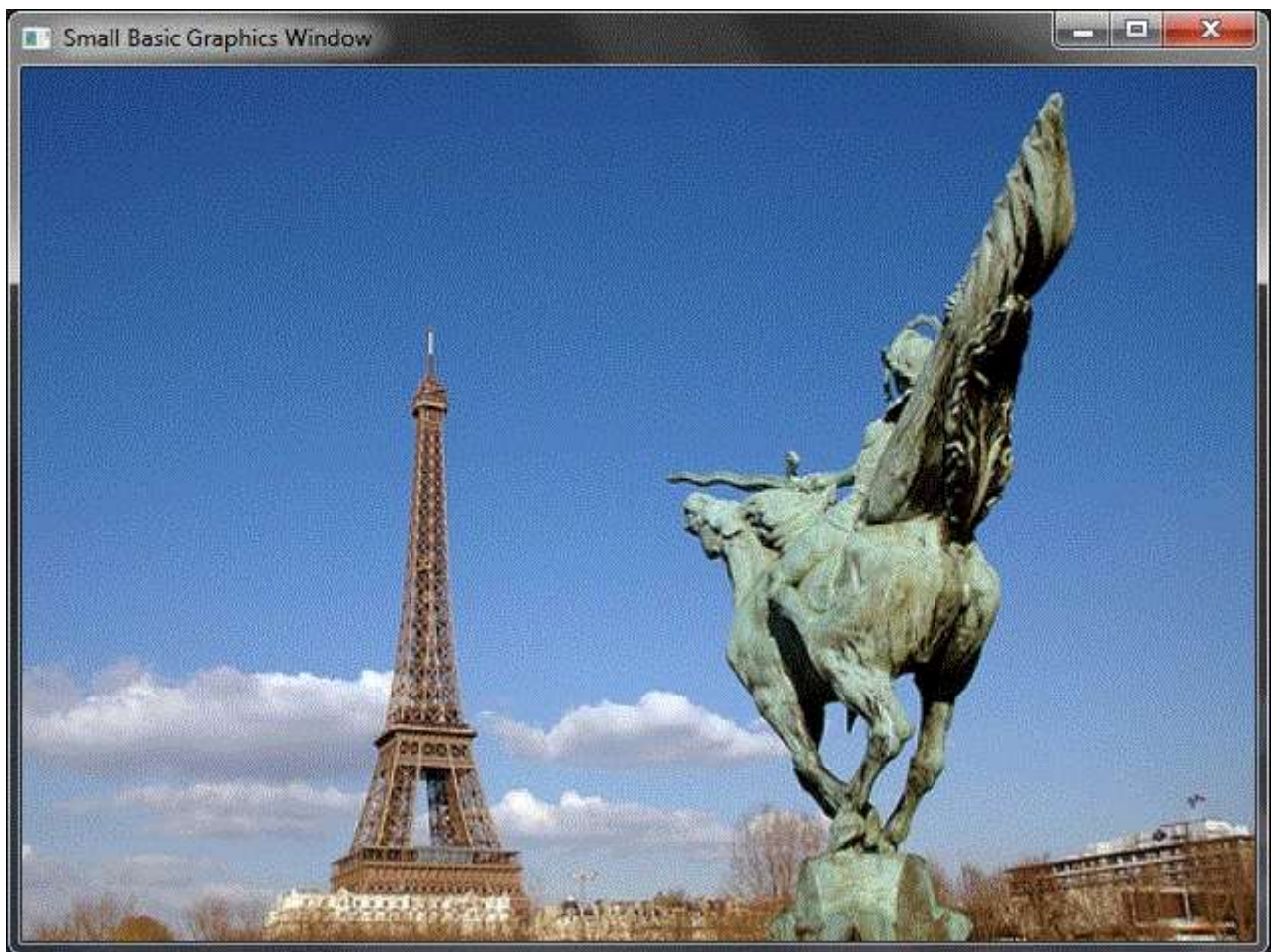
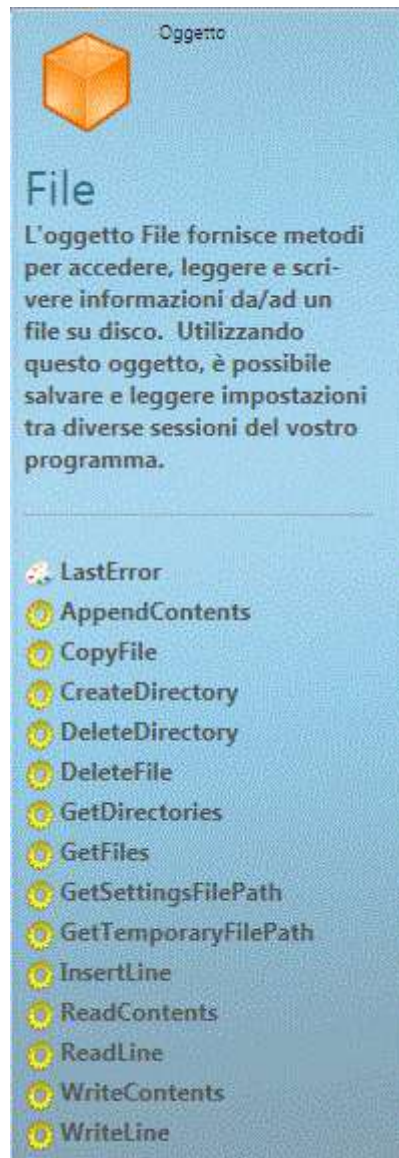
```
Orologio
Vuoi conoscere data e ora? s

Data                        03/04/2012
Anno                        2012
Mese                        4
Giorno                      3
Giorno della settimana      martedì
Orario                      18:23:31
Ora                         18
Minuti                      23
Secondi                     31
Millisecondi                415

Vuoi conoscere ancora data e ora? n
```

Esempio.

```
GraphicsWindow.BackgroundColor = "Aquamarine"
GraphicsWindow.Title = "Orologio"
GraphicsWindow.Width = 290
GraphicsWindow.Height = 90
GraphicsWindow.Show()
GraphicsWindow.Left = (Desktop.Width / 2) - (GraphicsWindow.Width / 2)
GraphicsWindow.Top = (Desktop.Height / 2) - (GraphicsWindow.Height / 2)
GraphicsWindow.CanResize = "False"
GraphicsWindow.BrushColor = "Black"
GraphicsWindow.FontName = "Arial New"
Timer.Interval = 1000
Timer.Tick = OnTick
Sub OnTick
  GraphicsWindow.Clear()
  GraphicsWindow.FontSize = 50
  If Clock.Hour > 12 Then
    hour = Clock.Hour - 12
    amPM = "PM"
  Else
    hour = Clock.Hour
    amPM = "AM"
  EndIf
  If Clock.Minute > 9 Then
    If Clock.Second > 9 Then
      GraphicsWindow.DrawText(0, 0, hour + ":" + Clock.Minute + ":" + Clock.Second + " " +
amPM)
    Else
      GraphicsWindow.DrawText(0, 0, hour + ":" + Clock.Minute + ":0" + Clock.Second + " "
+ amPM)
    EndIf
  Else
    If Clock.Second > 9 Then
      GraphicsWindow.DrawText(0, 0, hour + ":0" + Clock.Minute + ":" + Clock.Second + " "
+ amPM)
    Else
      GraphicsWindow.DrawText(0, 0, hour + ":0" + Clock.Minute + ":0" + Clock.Second + " "
+ amPM)
    EndIf
  EndIf
  GraphicsWindow.FontSize = 30
```

```
  GraphicsWindow.DrawText(0, 50, Clock.WeekDay + " " + Clock.Date)
EndSub
```

FLICKR

Esempio.

```
GraphicsWindow.BackgroundColor = "Black"
GraphicsWindow.MouseDown = OnMouseDown
Sub OnMouseDown
  pic = Flickr.GetRandomPicture("paris")
  GraphicsWindow.DrawResizedImage(pic, 0, 0, 640, 480)
EndSub
```

FILE

Esempio.

```
presenta()
Controlli()
Controls.ButtonClicked = Push
Sub Push
If Controls.LastClickedButton = ElimCar Then
  eliminaCartella()
EndIf
If Controls.LastClickedButton = Open then
  leggi()
EndIf
If Controls.LastClickedButton = BtnRec then
  Scrivi()
EndIf
If Controls.LastClickedButton = Mod Then
  modifica()
Endif
If Controls.LastClickedButton = Canc Then
```

```
  cancella()
EndIf
If Controls.LastClickedButton = Creac Then
  CreaCartella()
EndIf
If Controls.LastClickedButton = CopyBtn then
  CopiaFile()
EndIf
If Controls.LastClickedButton = Chiudi Then
    Program.End()
EndIf
Endsub
Sub Scrivi
  path = Controls.GetTextBoxText(NomeFile)
  txt = Controls.GetTextBoxText(testo)
  If File.AppendContents(path,txt) = "SUCCESS" then
    GraphicsWindow.ShowMessage("Scrittura effettuata con successo","Successo")
    ElseIf File.AppendContents(path,txt) = "FAILED" then
    GraphicsWindow.ShowMessage("Scrittura non effettuata","Errore")
  Endif
EndSub
Sub leggi
  path = Controls.GetTextBoxText(NomeFile)
  letto = File.ReadContents(path)
  Controls.SetTextBoxText(contenuto,letto)
EndSub
Sub modifica
  path = Controls.GetTextBoxText(NomeFile)
  New = Controls.GetTextBoxText(contenuto)
  If File.WriteContents(path,New) = "SUCCESS" then
    GraphicsWindow.ShowMessage("Modifica effettuata con successo","Successo")
    ElseIf File.WriteContents(path,New) = "FAILED" then
    GraphicsWindow.ShowMessage("Modifica non effettuata","Errore")
  Endif
Endsub
Sub eliminaCartella
    path = Controls.GetTextBoxText(CreateC)
    If File.DeleteDirectory(path) = "SUCCESS" then
    GraphicsWindow.ShowMessage("Eliminazione cartella effettuata con
successo","Successo")
    ElseIf File.DeleteDirectory(path) = "FAILED" then
    GraphicsWindow.ShowMessage("Eliminazione cartella non effettuata","Errore")
    Endif
  EndSub
Sub cancella
  path = Controls.GetTextBoxText(NomeFile)
  If File.DeleteFile(path) = "SUCCESS" then
    GraphicsWindow.ShowMessage("Cancellazione effettuata con successo","Successo")
    ElseIf File.DeleteFile(path) = "FAILED" then
    GraphicsWindow.ShowMessage("Cancellazione non effettuata","Errore")
  Endif
EndSub
Sub CreaCartella
  path = Controls.GetTextBoxText(CreateC)
```

```
  If File.CreateDirectory(path) = "SUCCESS" then
    GraphicsWindow.ShowMessage("Creazione cartella effettuata con
successo","Successo")
    ElseIf File.CreateDirectory(path) = "FAILED" then
    GraphicsWindow.ShowMessage("Creazione cartella non effettuata","Errore")
  Endif
EndSub
Sub CopiaFile
  path1 = Controls.GetTextBoxText(NomeFile)
  path2 = Controls.GetTextBoxText(CopyF)
    If File.CopyFile(path1,path2) = "SUCCESS" then
      GraphicsWindow.ShowMessage("Copia del file effettuata con successo","Successo")
    ElseIf File.CopyFile(path1,path2) = "FAILED" then
      GraphicsWindow.ShowMessage("Copia del file non effettuata","Errore")
    Endif
EndSub
Sub presenta
  GraphicsWindow.Width = 680
  GraphicsWindow.Height = 630
  GraphicsWindow.Title = "File di Testo"
  GraphicsWindow.BackgroundColor = "Aquamarine"
  GraphicsWindow.FontName = "Arial"
  GraphicsWindow.FontSize = 18
  GraphicsWindow.BrushColor = "Black"
EndSub
Sub Controlli
  GraphicsWindow.DrawText(195,25,"Inserisci percorso file su cui lavorare")
  GraphicsWindow.DrawText(195,75,"Inserisci percorso in cui copiare il file")
  GraphicsWindow.DrawText(195,125,"Inserisci percorso in cui creare cartella")
  CopyBtn = Controls.AddButton("Copia File",550,70)
  ElimCar = Controls.AddButton("Elimina cartella",270,170)
  Creac = Controls.AddButton("Crea Cartella",550,120)
  Chiudi = Controls.AddButton("X",650,590)
  Mod = Controls.AddButton("Modifica",565,170)
  Canc = Controls.AddButton("Cancella File",275,570)
  BtnRec = Controls.AddButton("Registra dati",115,570)
  Open = Controls.AddButton("Visualizza dati",435,570)
  NomeFile = Controls.AddTextBox(25,23)
  CopyF = Controls.AddTextBox(25,73)
  CreateC = Controls.AddTextBox(25,123)
  testo = Controls.AddMultiLineTextBox(25,210)
  contenuto = Controls.AddMultiLineTextBox(355,210)
  Controls.SetSize(testo,300,350)
  Controls.SetSize(contenuto,300,350)
EndSub
```

File di Testo
F:\Esempi SB\1.tx
Inserisci percorso file su cui lavorare
F:\Esempi SB\2
Inserisci percorso in cui copiare il file
Copia File
F:\Esempi SB\2
Inserisci percorso in cui creare cartella
Crea Cartella
Elimina cartella
Modifica
Utilizzo di Variabili.
Puoi accede ed utilizzare qualunque variabile presente in un'applicazione ll'interno di una routine.
Come esempio, la seguente applicazione accetta due numeri e stampa il piu' grande dei due.
Si noti che la variabile max e' utilizzata sia dentro che fuori della routine.
Utilizzo di Variabili.
Puoi accede ed utilizzare qualunque variabile presente in un'applicazione ll'interno di una routine.
Come esempio, la seguente applicazione accetta due numeri e stampa il piu' grande dei due.
Si noti che la variabile max e' utilizzata sia dentro che fuori della routine.
Registra dati
Cancella File
Visualizza dati
X

LOGO

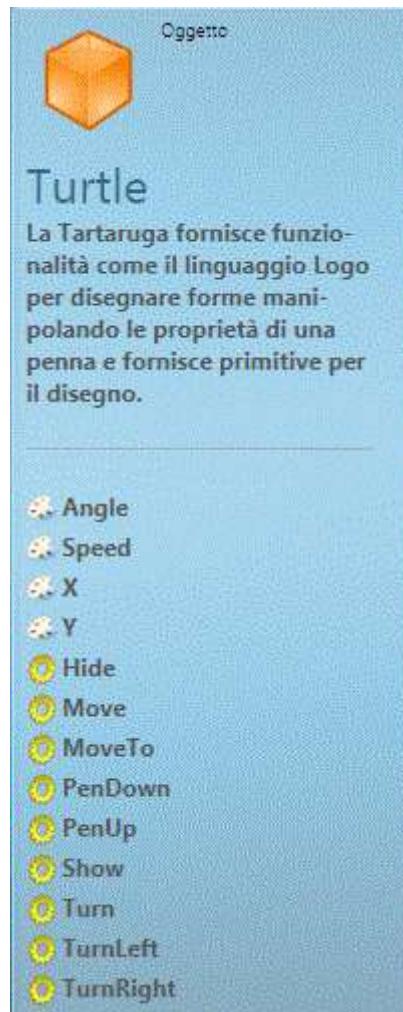
INTRODUZIONE

Dal greco logos “parola”, progettato da Brown e Feurzeig presso la Bolt di Cambridge, Massachussets.

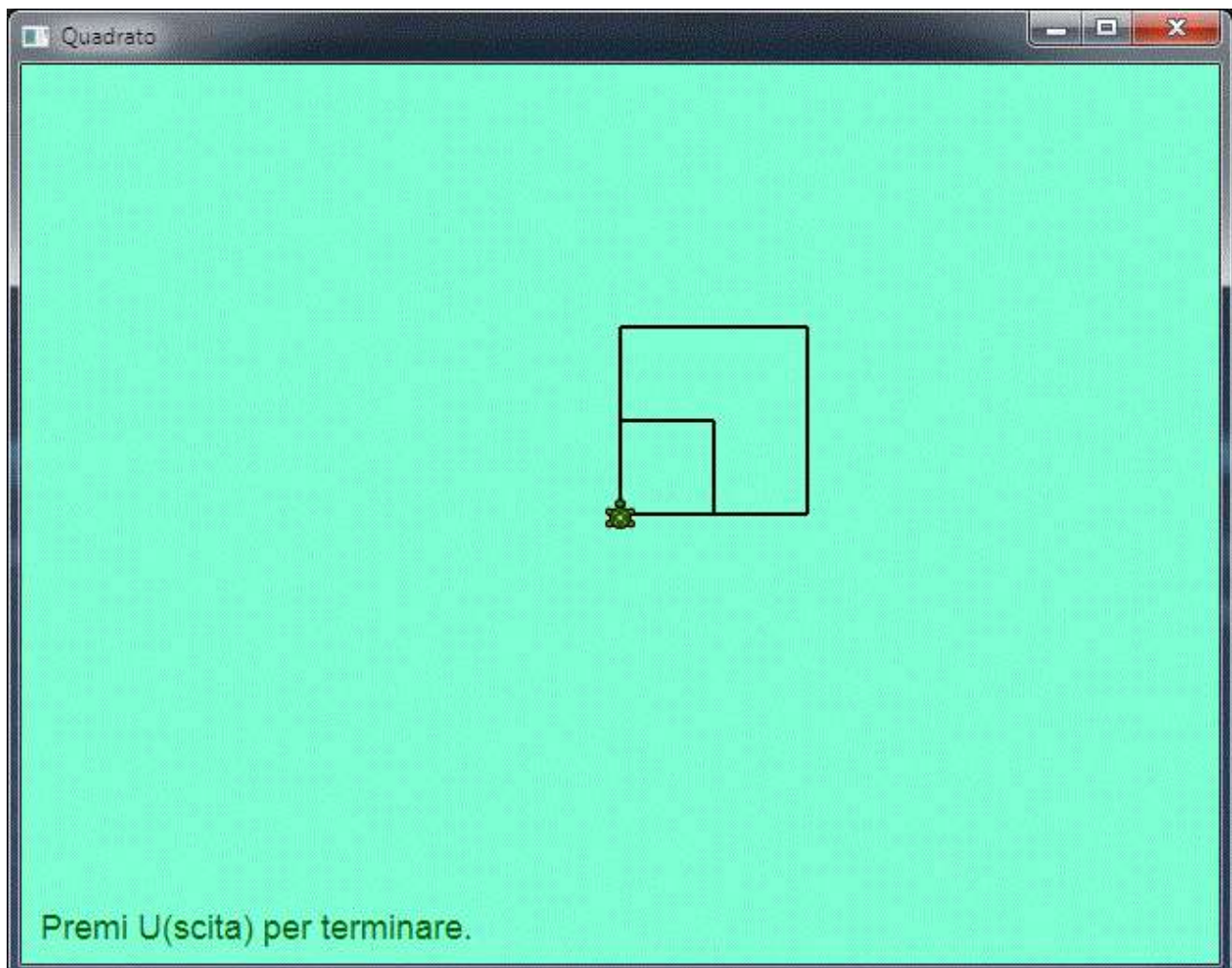
Punto d’incontro delle teorie di Jean Piaget (Neuchâtel, 9 agosto 1896 – Ginevra, 16 settembre 1980) sulla genesi del pensiero e lo sviluppo intellettuale combinate con la ricerca sull’intelligenza artificiale condotta da Seymour Papert (Pretoria, 1 marzo 1928) e Marvin Lee Minsky (New York, 9 agosto 1927) al **MIT** *(Massachusetts Institute of Technology*).
Caratteristica è la grafica con la “tartaruga”, che permette al programmatore di realizzare semplici disegni ordinando ad una tartaruga, una sorta di cursore, di muoversi sullo schermo nelle diverse direzioni, derivata dal linguaggio **LISP** (*LISt Processing*).
È un automa che apprende perché le routine sono viste come nuove primitive.

“Che siano i bambini a programmare il calcolatore e non il calcolatore a programmare i bambini.” (Seymour Papert).

In Small Basic la tartaruga si è evoluta graficamente, ma permette di fare sostanzialmente le stesse cose della tartaruga di LOGO.

TURTLE

Esempio, quadrati.

```
GraphicsWindow.KeyDown = OnKeyDown
presenta()
quadrato()
quadratoI()
stampa()
Sub quadrato
	Turtle.Show()
	Turtle.Move(50)
	Turtle.TurnRight()
	Turtle.Move(50)
	Turtle.TurnRight()
	Turtle.Move(50)
	Turtle.TurnRight()
	Turtle.Move(50)
	Turtle.TurnRight()
EndSub
Sub quadratoI
	For i=1 To 4
		Turtle.Move(100)
		Turtle.TurnRight()
	EndFor
```

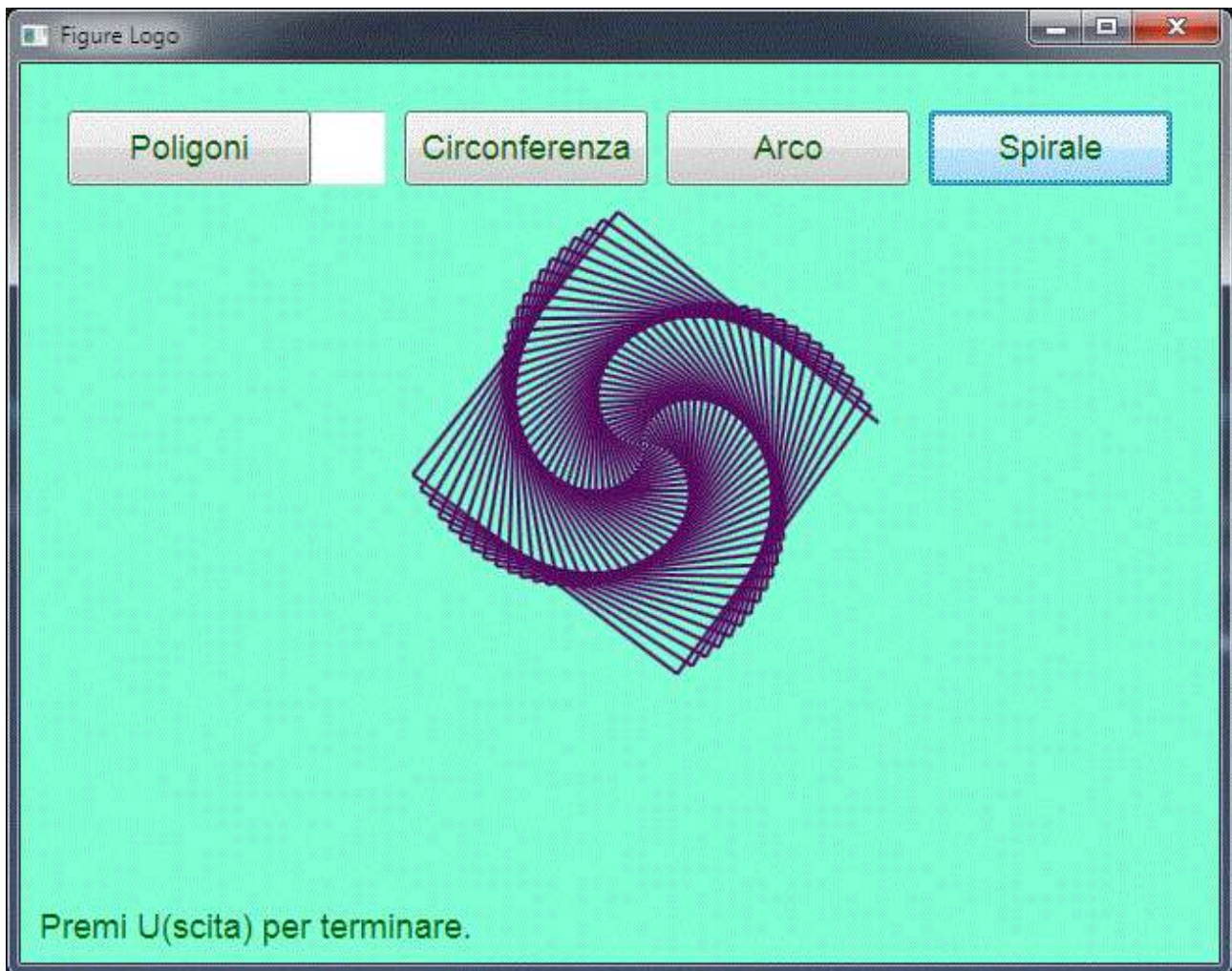
```
EndSub
Sub presenta
	GraphicsWindow.BackgroundColor = "Aquamarine"
	GraphicsWindow.Title = "Quadrato"
	GraphicsWindow.Width = 640
	GraphicsWindow.Height = 480
	GraphicsWindow.Show()
	GraphicsWindow.Clear()
	' mostriamo la tartaruga
	Turtle.Show()
	' velocità della tartaruga tra 1 e 10
	Turtle.Speed = 5
EndSub
Sub stampa
	GraphicsWindow.FontBold = "False"
	GraphicsWindow.FontName = "Arial"
	GraphicsWindow.FontSize = 18
	GraphicsWindow.BrushColor = "DarkGreen"
	GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub OnKeyDown
	If (GraphicsWindow.LastKey = "U") Then
		Program.End()
	Endif
EndSub
```

Esempio, poligoni.

```
GraphicsWindow.KeyDown = OnKeyDown
presenta()
poligoni()
stampa()
Sub poligoni
	' numero arbitrario di lati
	' per i poligoni, i cui lati sono tutti uguali, l'angolo tra i lati è ottenuto
	'  dividendo 360 per il numero dei lati
	' modificare la variabile lati per ottenere un poligono qualunque
	' per esempio con lati = 4 si ottiene un quadrato
	lati = 6
	lun = 400/lati
	angolo = 360/lati
	For i=1 To lati
		For j = 1 To lati
			Turtle.Move(lun/12)
			Turtle.PenUp ()
			Turtle.Move(lun/12)
			Turtle.PenDown ()
		EndFor
		Turtle.Turn(angolo)
	EndFor
EndSub
Sub presenta
	GraphicsWindow.BackgroundColor = "Aquamarine"
	GraphicsWindow.Title = "Poligoni"
	GraphicsWindow.Width = 640
	GraphicsWindow.Height = 480
	GraphicsWindow.Show()
	GraphicsWindow.Clear()
	' mostriamo la tartaruga
	Turtle.Show()
	' velocità della tartaruga tra 1 e 10
	Turtle.Speed = 5
EndSub
Sub stampa
	GraphicsWindow.FontBold = "False"
	GraphicsWindow.FontName = "Arial"
	GraphicsWindow.FontSize = 18
	GraphicsWindow.BrushColor = "DarkGreen"
	GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub OnKeyDown
	If (GraphicsWindow.LastKey = "U") Then
		Program.End()
	Endif
EndSub
```

Esempio, figure.

```
GraphicsWindow.KeyDown = OnKeyDown
Controls.ButtonClicked = BC
presenta()
begin:
GraphicsWindow.Clear()
stampa()
bottoni()
Program.Delay(5000)
Sub BC
If Controls.LastClickedButton = Pol then
  lati = Controls.GetTextBoxText(TxtPol)
  If lati < 3 then
   Sound.PlayBellRing()
   GraphicsWindow.ShowMessage("Inserimento lati errato!","Errore")
  else
  poligoni()
  endif
endif
If Controls.LastClickedButton = spi then
  GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
  spirale()
endif
If Controls.LastClickedButton = cir then
```

```
  GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
  circonferenza()
endif
If Controls.LastClickedButton = arc then
  GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
  arco()
  endif
EndSub
Program.Delay(10000)
Goto begin
Sub poligoni
  ' numero arbitrario di lati
  ' per i poligoni, i cui lati sono tutti uguali,  l'angolo tra i lati è ottenuto
  '  dividendo 360 per il numero dei lati
  ' modificare la variabile lati per ottenere un poligono qualunque
  ' per esempio con lati = 4 si ottiene un quadrato
  lun = 400/lati
  angolo = 360/lati
  For i = 1 to lati
    GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
    Turtle.Move(lun)
    Turtle.Turn(angolo)
  EndFor
EndSub
Sub circonferenza
  For i = 1 to 360
     Turtle.Move(1)
    Turtle.Turn(1)
  EndFor
EndSub
Sub arco
  gradi = 40
  For i = 1 to gradi
    Turtle.Move(1)
    Turtle.Turn(1)
  EndFor
EndSub
Sub spirale
 side = 0
 angle = 91
 For i = 1 to 180
  Turtle.Move(side)
  Turtle.Turn(angle)
  side = side + 1
 EndFor
EndSub
Sub presenta
  GraphicsWindow.BackgroundColor = "Aquamarine"
  GraphicsWindow.Title = "Figure Logo"
  GraphicsWindow.Width = 640
  GraphicsWindow.Height = 480
  GraphicsWindow.Show()
  GraphicsWindow.Clear()
  ' mostriamo la tartaruga
```

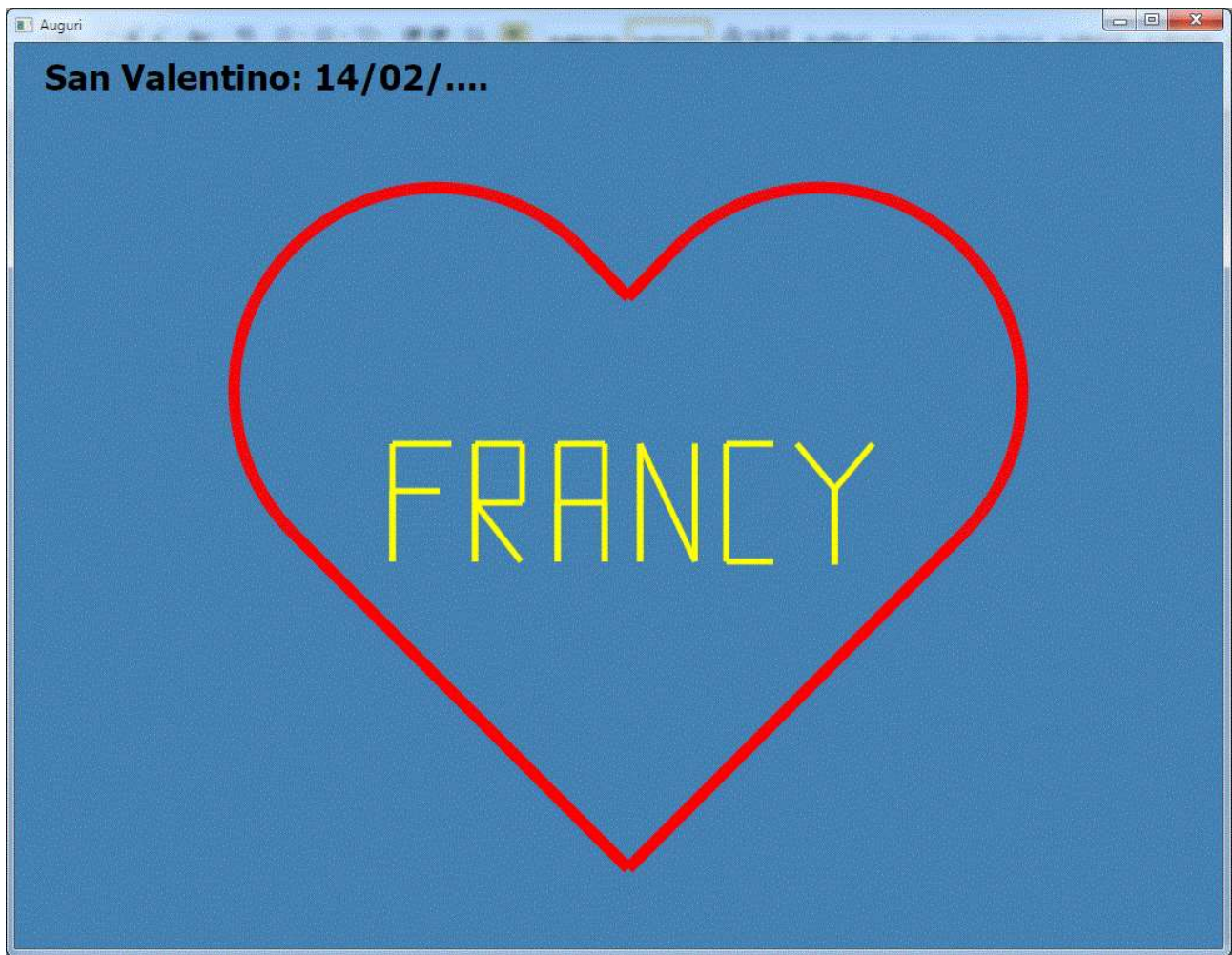
```
 Turtle.Show()
 ' velocità della tartaruga tra 1 e 10
 Turtle.Speed = 10
EndSub
Sub stampa
 GraphicsWindow.FontBold = "False"
 GraphicsWindow.FontName = "Arial"
 GraphicsWindow.FontSize = 18
 GraphicsWindow.BrushColor = "DarkGreen"
 GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub bottoni
 Pol = Controls.AddButton("Poligoni",25,25)
 TxtPol = Controls.AddTextBox(155,25)
 cir = Controls.AddButton("Circonferenza",205,25)
 arc = Controls.AddButton("Arco",345,25)
 spi = Controls.AddButton("Spirale",485,25)
 Controls.SetSize(cir,130,40)
 Controls.SetSize(Pol,130,40)
 Controls.SetSize(arc,130,40)
 Controls.SetSize(spi,130,40)
 Controls.SetSize(TxtPol,40,40)
EndSub
Sub OnKeyDown
 If (GraphicsWindow.LastKey = "U") Then
  Program.End()
 endif
EndSub
```

Esempio, auguri.

```
Sound.Play ("F:\Esempi SB\felicita.mp3")
presenta()
Turtle.PenUp()
Turtle.Turn(180)
Turtle.Move(100)
Turtle.PenDown()
GraphicsWindow.PenColor="Yellow"
GraphicsWindow.PenWidth = 5
letteraeffe()
letteraerre()
letteraa()
letteraenne()
letterac()
letteray()
GraphicsWindow.PenColor= "Red"
GraphicsWindow.PenWidth = 10
cuore()
Sub letteraeffe
  Turtle.Move(100)
  Turtle.Turn(180)
  Turtle. Move(60)
  Turtle.Turn(90)
  Turtle.Move(40)
```

```
  Turtle.Penup()
  Turtle.Move(10)
  Turtle.Turn(-90)
  Turtle.Move(40)
  Turtle.Turn(-90)
  Turtle.PenDown()
  Turtle.Move(50)
EndSub
Sub letteraerre
  Turtle.Turn(180)
  Turtle.PenUp()
  Turtle.Move(70)
  Turtle.Turn(90)
  Turtle.PenDown()
  a = Turtle.x
  b = Turtle.y
  Turtle.Move(100)
  Turtle.Turn(180)
  Turtle.Move(50)
  Turtle.Turn(90)
  Turtle.Move(40)
  Turtle.Turn(-90)
  Turtle.Move(50)
  Turtle.Turn(-90)
  Turtle.Move(40)
  Turtle.Turn(-90)
  Turtle.Move(50)
  Turtle.Turn(-38)
  Turtle.Move(63)
EndSub
Sub letteraa
  Turtle.Turn(38)
  Turtle.Turn(180)
  Turtle.X = a
  Turtle.Y = b
  Turtle.PenUp()
  turtle.TurnRight()
  Turtle.Move(70)
  Turtle.TurnRight()
  Turtle.PenDown()
  Turtle.Move(100)
  Turtle.Turn(180)
  Turtle.Move(50)
  Turtle.TurnRight()
  Turtle.Move(40)
  Turtle.TurnRight()
  Turtle.Move(50)
  Turtle.Turn(180)
  Turtle.Move(100)
  Turtle.TurnLeft()
  Turtle.Move(40)
  Turtle.Turn(180)
  Turtle.PenUp()
  Turtle.Move(70)
```

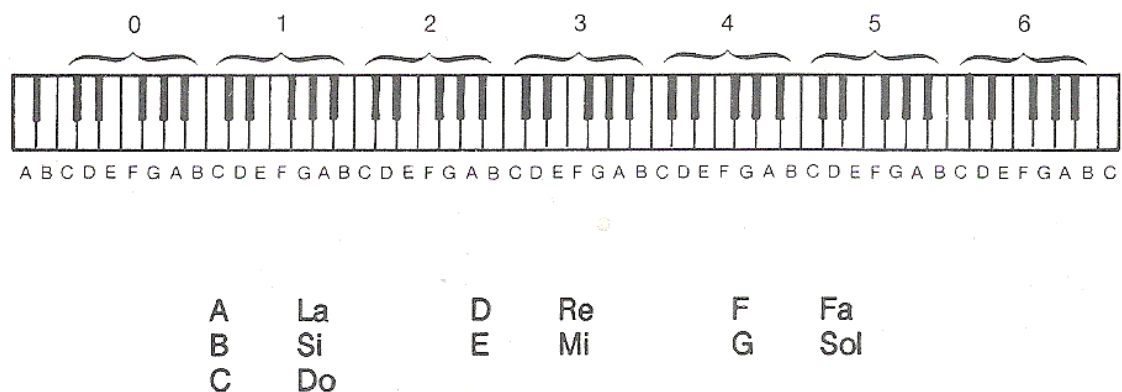
```
EndSub
Sub letteraenne
  Turtle.TurnRight()
  Turtle.PenDown()
  a = Turtle.x
  b = Turtle.y
  Turtle.Move(100)
  Turtle.Turn(180)
  Turtle.Move(100)
  Turtle.Turn(180)
  Turtle.Turn(-25)
  Turtle.Move(110)
  Turtle.Angle = 0
  Turtle.Move(100)
  Turtle.X = a
  Turtle.Y = b
EndSub
Sub letterac
  Turtle.TurnRight()
  Turtle.PenUp()
  Turtle.Move(73)
  Turtle.TurnRight()
  Turtle.PenDown()
  Turtle.Move(100)
  Turtle.TurnLeft()
  Turtle.Move(40)
  Turtle.TurnLeft()
  Turtle.PenUp()
  Turtle.Move(100)
  Turtle.TurnLeft()
  turtle.PenDown()
  Turtle.Move(40)
  Turtle.Turn(180)
  Turtle.PenUp()
  Turtle.Move(60)
EndSub
Sub letteray
  Turtle.PenDown()
  Turtle.Turn(50)
  Turtle.Move(50)
  Turtle.Turn(40)
  Turtle.Move(64)
  Turtle.Turn(180)
  Turtle.Move(64)
  Turtle.Turn(40)
  Turtle.Move(50)
  Turtle.Turn(50)
EndSub
Sub cuore
  Turtle.Angle = 0
  Turtle.X = 520
  Turtle.Y = 700
  Turtle.Turn(-45)
  Turtle.Move(400)
```

```
  For i = 0 To 180
    Turtle.Move(3)
    Turtle.Turn(1)
  endfor
  Turtle.Move(58)
  Turtle.Angle = 0
  Turtle.X = 520
  Turtle.Y = 700
  Turtle.Turn(45)
  Turtle.Move(400)
    For i = 0 To 180
    Turtle.Move(3)
    Turtle.Turn(-1)
  endfor
  Turtle.Move(58)
  Turtle.PenUp()
  Turtle.Move(1000)
EndSub
Sub presenta
  GraphicsWindow.BackgroundColor = "SteelBlue"
  GraphicsWindow.Title = "Auguri"
  GraphicsWindow.Width = 1024
  GraphicsWindow.Height = 768
  GraphicsWindow.Show()
  GraphicsWindow.Clear()
  GraphicsWindow.BrushColor ="Black"
  GraphicsWindow.FontSize="30"
  GraphicsWindow.DrawBoundText(25,10,600,"San Valentino: 14/02/....")
  ' mostriamo la tartaruga
  Turtle.Show()
  ' velocità della tartaruga tra 1 e 10
  Turtle.Speed = 7
EndSub
```

Auguri
San Valentino: 14/02/....
FRANCY

MUSICA

SOUND

SOUND.PLAYMUSIC(NOTES)

Dove *notes* è un insieme di note musicali da suonare, non restituisce nulla.
Gli effetti sonori si ottengono specificando l'altezza e la durata dei suoni, il tempo e l'eventuale sospensione dell'elaborazione dell'applicazione durante l'esecuzione sonora.
Il PC può eseguire 84 delle 88 note della scala temperata disponibili sulla tastiera di un pianoforte, per un'estensione di 7 ottave.

Le note s'individuano nel modo seguente.

Nome della nota

Notazione anglosassone	Notazione latina
A	LA
B	SI
C	DO
D	RE
E	MI
F	FA
G	SOL

Ottava
La lettera O seguita da un intero da 0 a 6, dal grave all'acuto, individua l'ottava per tutte le note seguenti, fino al prossimo cambio di ottava; se non s'indica alcuna ottava, vale per default la quarta.
Per salire o scendere di una sola ottava o anche per impostarla inizialmente, si può usare < o >.
È possibile utilizzarne anche più di uno per volta, ad esempio per salire di due o più ottave.

Esempio.
```
Sound.Playmusic("O4 B A C G < G F B C >>> A F G")
```

Alterazioni
Si designano in questo modo.
Le note di DIESIS sono prodotte aggiungendo + oppure # dopo le note normali.

Esempio.
```
Sound.PlayMusic("G#")     ' SOL DIESIS equivale G#
```

Le note di BEMOLLE sono prodotte aggiungendo - dopo le note normali.

Esempio.
```
Sound.PlayMusic("B-")     ' SI BEMOLLE equivale B-
```

Esempi.
```
' emette il DO all'ottava centrale
Sound.PlayMusic("O3 C")

' si emettono 4 note in successione
' le prime due della seconda ottava RE MI
' le ultime due della quarta ottava FA DIESIS e SOL DIESIS
Sound.PlayMusic("O2 DE O4 F+G+")

' note su diverse ottave
' MI e FA DIESIS dell'ottava centrale
' DO, RE BEMOLLE e MI della prima ottava
' MI e FA DIESIS
Sound.PlayMusic("EF+ O1 CD-E EF#")
```

```
' suona la scala completa di RE minore
Sound.PlayMusic("D E F G A B- O5 C+ D")

' tanti auguri a te
Sound.PlayMusic("CCDCFE")
```

Durata
Le note si distinguono nel modo seguente.
- ✓ Interi (1).
- ✓ Mezzi (2), la nota è riprodotta con durata dimezzata.
- ✓ Quarti (4): valore di default.
- ✓ Ottavi (8).
- ✓ Sedicesimi (16).
- ✓ Trentaduesimi (32).
- ✓ Sessantaquattresimi (64).

La durata è specificata da un numero che segue la nota alla quale applicarla.

Esempio.
```
' suona il DO tre volte
' la prima come intero
' la seconda come un mezzo
' la terza come un quarto
Sound.PlayMusic("C1 C2 C4")
```

Con la lettera P seguita da un intero *n* compreso tra 1 e 64 si riproduce una pausa di durata *1/n*.

Esempio.
```
' suona il DO tre volte
' con una prima pausa di 1/2
' una seconda pausa di 1/16
Sound.PlayMusic("C P2 C P16 C")
```

Si riesce a eseguire l'aumento di metà durata, comune nella pratica musicale, basta far seguire il nome della nota dal carattere punto.

```
' suona il DO sei volte con durata
' 1/4, 3/8, 3/16, 9/16, 3/4 e 1/10
' con una  pausa di 3/8 tra la quarta e quinta nota
Sound.PlayMusic("C C. C8. C.. P4 C2. C")
```

Tempi musicali

Tempo	Quarti al minuto
Largo	40-60 BPM
Larghetto	60-66 BPM
Adagio	66-76 BPM
Andante	76-108 BPM
Moderato	108-120 BPM
Allegro	120-168 BPM
Presto	168-200 BPM
Prestissimo	200-208 BPM

Con la lettera T seguita da un intero *n* compreso tra 32 e 255 si ottiene l'esecuzione al tempo di *n* quarti al minuto, l'assegnazione vale fino a diversa indicazione e se non è data vale il tempo di 120 quarti al minuto.

Esempio.

```
' suona il DO due volte
' la prima volta per 1/2 secondo
' la seconda volta per 1 secondo
Sound.PlayMusic("C T60 C")
```

Le note possono essere individuate anche con un numero progressivo, secondo la tabella seguente, da indicare dopo la lettera N.

						Ottava							
0		1		2		3		4		5		6	
1	Do	13	Do	25	Do	37	Do	49	Do	61	Do	73	Do
2	Do#	14	Do#	26	Do#	38	Do#	50	Do#	62	Do#	74	Do#
3	Re	15	Re	27	Re	39	Re	51	Re	63	Re	75	Re
4	Re#	16	Re#	28	Re#	40	Re#	52	Re#	64	Re#	76	Re#
5	Mi	17	Mi	29	Mi	41	Mi	53	Mi	65	Mi	77	Mi
6	Fa	18	Fa	30	Fa	42	Fa	54	Fa	66	Fa	78	Fa
7	Fa#	19	Fa#	31	Fa#	43	Fa#	55	Fa#	67	Fa#	79	Fa#
8	Sol	20	Sol	32	Sol	44	Sol	56	Sol	68	Sol	80	Sol
9	Sol#	21	Sol#	33	Sol#	45	Sol#	57	Sol#	69	Sol#	81	Sol#
10	La	22	La	34	La	46	La	58	La	70	La	82	La
11	La#	23	La#	35	La#	47	La#	59	La#	71	La#	83	La#
12	Si	24	Si	36	Si	48	Si	60	Si	72	Si	84	Si

Esempio.

```
Sound.PlayMusic("N1 N2")
```

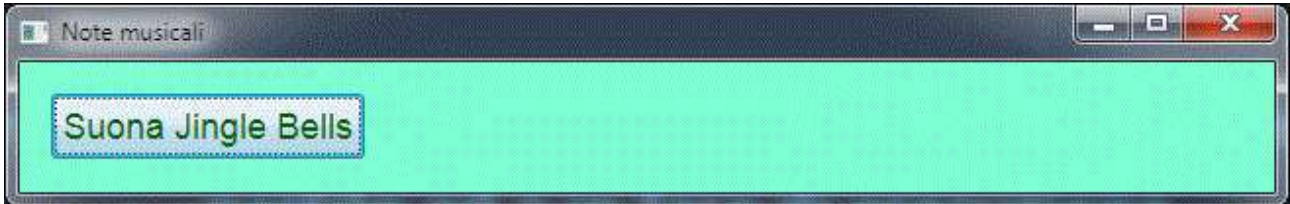
Esempio.

```
GraphicsWindow.KeyDown = OnKeyDown
presenta()
stampa()
bet = Controls.AddButton("Suona Per Elisa",16,16)
Controls.ButtonClicked =suona
Sub suona
  If Controls.LastClickedButton = bet Then
     perelisa()
  EndIf
endsub
Sub perelisa
  Sound.PlayMusic("e d# e d# e b d c a c e a b e g# b c")
  Sound.PlayMusic(" e d# e d# e b d c a c e a b e c b a")
  Sound.PlayMusic(" b c d e g f e d f e d c e d c b")
  Sound.PlayMusic("e d# e d# e b d c a c e a b e g# b c")
Endsub
Sub presenta
  GraphicsWindow.BackgroundColor = "Aquamarine"
  GraphicsWindow.Title = "Note musicali"
  GraphicsWindow.Width = 640
  GraphicsWindow.Height = 480
```

```
  GraphicsWindow.Show()
  GraphicsWindow.Clear()
  GraphicsWindow.BrushColor = "Black"
EndSub
Sub stampa
  GraphicsWindow.FontBold = "False"
  GraphicsWindow.FontName = "Arial"
  GraphicsWindow.FontSize = 18
  GraphicsWindow.BrushColor = "Red"
  GraphicsWindow.BrushColor = "DarkGreen"
  GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub OnKeyDown
  If (GraphicsWindow.LastKey = "U") Then
    Program.End()
  Endif
EndSub
```

Esempio.

```
GraphicsWindow.KeyDown = OnKeyDown
presenta()
stampa()
bet = Controls.AddButton("Suona Jingle Bells",16,16)
Controls.ButtonClicked =suona
Sub suona
  If Controls.LastClickedButton = bet Then
      Jingle()
  EndIf
endsub
Sub Jingle
    Sound.PlayMusic("e8 e8 e e8 e8 e e8 g8 c8 d8 e2 f8 f8 f f8 e8 e e8 d8 d8 e8 d g")
    Sound.PlayMusic("e8 e8 e e8 e8 e e8 g8 c8 d8 e2 f8 f8 f f8 e8 e g8 g8 f8 d8 c2")
EndSub
Sub presenta
  GraphicsWindow.BackgroundColor = "Aquamarine"
  GraphicsWindow.Title = "Note musicali"
  GraphicsWindow.Width = 640
  GraphicsWindow.Height = 480
  GraphicsWindow.Show()
  GraphicsWindow.Clear()
  GraphicsWindow.BrushColor = "Black"
EndSub
Sub stampa
  GraphicsWindow.FontBold = "False"
  GraphicsWindow.FontName = "Arial"
  GraphicsWindow.FontSize = 18
```

```
  GraphicsWindow.BrushColor = "Red"
  GraphicsWindow.BrushColor = "DarkGreen"
  GraphicsWindow.DrawBoundText(10,450,300,"Premi U(scita) per terminare.")
EndSub
Sub OnKeyDown
  If (GraphicsWindow.LastKey = "U") Then
    Program.End()
  Endif
EndSub
```

Esempio, simulare un pianoforte.

```
Sound.PlayChime()
GraphicsWindow.Title = "                              ?  Pianoforte  ?"
GraphicsWindow.BackgroundColor = "SlateGray"
GraphicsWindow.BrushColor = "Black"
GraphicsWindow.CanResize = "False"
Controls.ButtonClicked = BC
GraphicsWindow.Width  = 340
GraphicsWindow.Height = 200
GraphicsWindow.DrawBoundText(85, 20,400,"? Premi un Pulsante... ?")
toni()
semitoni()
Sub toni
  norm = GraphicsWindow.FontName
  GraphicsWindow.FontName = "WingDings"
  bell = Controls.AddButton("%", 315 ,12)
  GraphicsWindow.FontName = norm
  C = Controls.AddButton("DO",30 , 90)
  Controls.SetSize(C , 40 , 100)
  D = Controls.AddButton("RE",70 , 90)
  Controls.SetSize(D , 40 , 100)
  E = Controls.AddButton("MI",110, 90)
  Controls.SetSize(E , 40 , 100)
  F = Controls.AddButton("FA",150, 90)
  Controls.SetSize(F , 40 , 100)
  G = Controls.AddButton("SOL",190,90)
  Controls.SetSize(G , 40 , 100)
  A = Controls.AddButton("LA",230, 90)
  Controls.SetSize(A , 40 , 100)
  B = Controls.AddButton("SI",270, 90)
  Controls.SetSize(B , 40 , 100)
EndSub
Sub semitoni
  Cd = Controls.AddButton("DO#",50 ,50)
  Controls.SetSize(Cd, 40 , 80)
  Dd = Controls.AddButton("RE#",90 ,50)
```

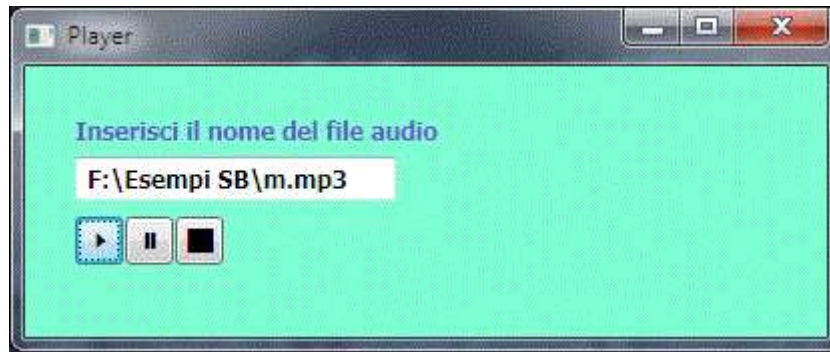
```
  Controls.SetSize(Dd, 40 , 80)
  Fd = Controls.AddButton("FA#",170,50)
  Controls.SetSize(Fd, 40 , 80)
  GraphicsWindow.FontSize = GraphicsWindow.FontSize -1.6
  Gd = Controls.AddButton("SOL#",210,50)
  GraphicsWindow.FontSize = GraphicsWindow.FontSize +1.6
  Controls.SetSize(Gd, 40 , 80)
  Ad = Controls.AddButton("LA#",250,50)
  Controls.SetSize(Ad, 40 , 80)
EndSub
Sub BC
    If ( Controls.LastClickedButton = bell)Then
      Sound.PlayBellRing()
      EndIf
    If ( Controls.LastClickedButton = C)Then
      Sound.PlayMusic("C4")
      EndIf
    If ( Controls.LastClickedButton = Cd)Then
      Sound.PlayMusic("C#4")
      EndIf
    If ( Controls.LastClickedButton = D)Then
      Sound.PlayMusic("D4")
      EndIf
    If ( Controls.LastClickedButton = Dd)Then
      Sound.PlayMusic("D#4")
      EndIf
    If ( Controls.LastClickedButton = E)Then
      Sound.PlayMusic("E4")
      EndIf
    If ( Controls.LastClickedButton = F)Then
      Sound.PlayMusic("F4")
      EndIf
    If ( Controls.LastClickedButton = Fd)Then
      Sound.PlayMusic("F#4")
      EndIf
    If ( Controls.LastClickedButton = G)Then
      Sound.PlayMusic("G4")
      EndIf
    If ( Controls.LastClickedButton = Gd)Then
      Sound.PlayMusic("G#4")
      EndIf
    If ( Controls.LastClickedButton = A)Then
      Sound.PlayMusic("A4")
      EndIf
    If ( Controls.LastClickedButton = Ad)Then
      Sound.PlayMusic("A#4")
      EndIf
    If ( Controls.LastClickedButton = B)Then
      Sound.PlayMusic("B4")
      EndIf
EndSub
```

SOUND.PLAY

Riproduce file audio: **MP3** (*Motion Picture Expert Group-1/2 Audio Layer 3*), un WAV (WAVE) o un **WMA** (*Windows Media Audio*), altri formati di file potrebbero essere riprodotti o meno a seconda dei **CODEC** (*COdificare e/o DECodificare*) installati sul PC.
Se il file è stato messo in pausa, quest'operazione riprenderà dalla posizione da cui la riproduzione era stata interrotta.

Esempio.

```
presenta()
GraphicsWindow.DrawText(25, 25, "Inserisci il nome del file audio")
GraphicsWindow.PenColor = "#000000"
GraphicsWindow.BrushColor = "#000000"
fn2enter = Controls.AddTextBox(25, 45)
GraphicsWindow.FontName = "Webdings"
bttnplay = Controls.AddButton("4", 25, 75)
bttnpause = Controls.AddButton(";", 50, 75)
bttnstop = Controls.AddButton("g", 75, 75)
begin:
Program.Delay(100)
If Controls.LastClickedButton = bttnplay Then
  filename = Controls.GetTextBoxText(fn2enter)
  filenameubo = Controls.GetTextBoxText(fn2enter)
  Sound.Play(filename)
EndIf
If Controls.LastClickedButton = bttnpause Then
  If filename = filenameubo Then
    Sound.Pause(filename)
  Else
    Sound.Pause(filename)
    Sound.Pause(filenameubo)
  EndIf
EndIf
If Controls.LastClickedButton = bttnstop Then
    If filename = filenameubo Then
    Sound.Stop(filename)
  Else
    Sound.Stop(filename)
    Sound.Stop(filenameubo)
```

```
  EndIf
EndIf
Goto begin
Sub presenta
  GraphicsWindow.BackgroundColor = "Aquamarine"
  GraphicsWindow.Title = "Player"
  GraphicsWindow.Width = 400
  GraphicsWindow.Height = 135
  GraphicsWindow.Show()
  GraphicsWindow.Clear()
EndSub
```

IMAGELIST

Esempio, progettare un lettore MP3.

```
presenta()
Sub presenta
background=ImageList.LoadImage("F:\Esempi SB\skinmp3.jpg")
GraphicsWindow.Title = "MP3 Player"
GraphicsWindow.DrawImage(background , -300 , -220)
GraphicsWindow.Top = 10
GraphicsWindow.Left= 10
GraphicsWindow.Height= 350
GraphicsWindow.Width = 400
GraphicsWindow.CanResize = "False"
Controls.ButtonClicked = BC
buttons()
EndSub
Sub buttons
  GraphicsWindow.BrushColor= "Black"
  Controls.AddButton("Inserisci il percorso del FILE audio" , 100 , 120 )
  Controls.AddButton(" --> ", 50 , 150 )
  Controls.AddButton(" <-- ", 333, 150 )
  path = Controls.AddTextBox( 120 , 150 )
  Controls.SetSize(path , 180 , 25)
  GraphicsWindow.FontName = "Webdings"
  Program.Delay(500)
  play= Controls.AddButton(" 4 " , 122.5 , 180)
  Program.Delay(500)
  pause=Controls.AddButton(" ; " , 192.5 , 180)
  Program.Delay(500)
  stop =Controls.AddButton(" g " , 262.5 , 180)
  GraphicsWindow.FontName = "Arial"
  GraphicsWindow.FontBold = "True"
  Program.Delay(1000)
  end  = Controls.AddButton("Click Here to End the Application", 190 , 315)
EndSub
Sub BC
  If ( Controls.LastClickedButton = play)Then
```

```
    filename = Controls.GetTextBoxText(path)
    Sound.Play(filename)
  EndIf
  If ( Controls.LastClickedButton = pause)Then
    Sound.Pause(filename)
  EndIf
  If( Controls.LastClickedButton = stop)Then
    Sound.Stop(filename)
  EndIf
  If( Controls.LastClickedButton = end)Then
    Program.End()
  EndIf
EndSub
```

PLUGIN

INTRODUZIONE

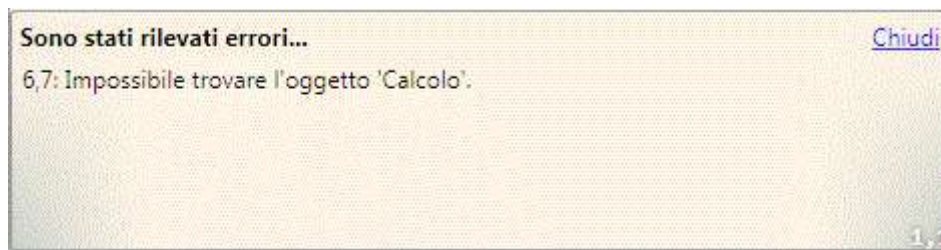
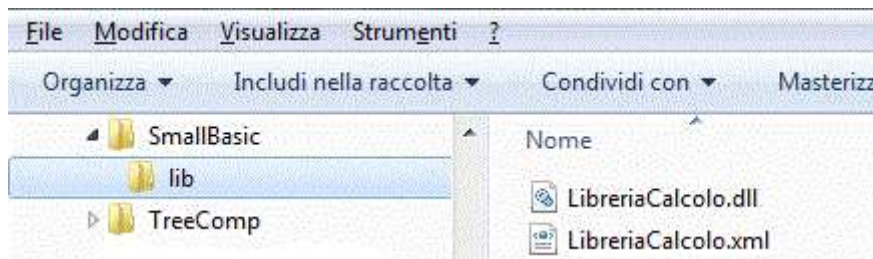
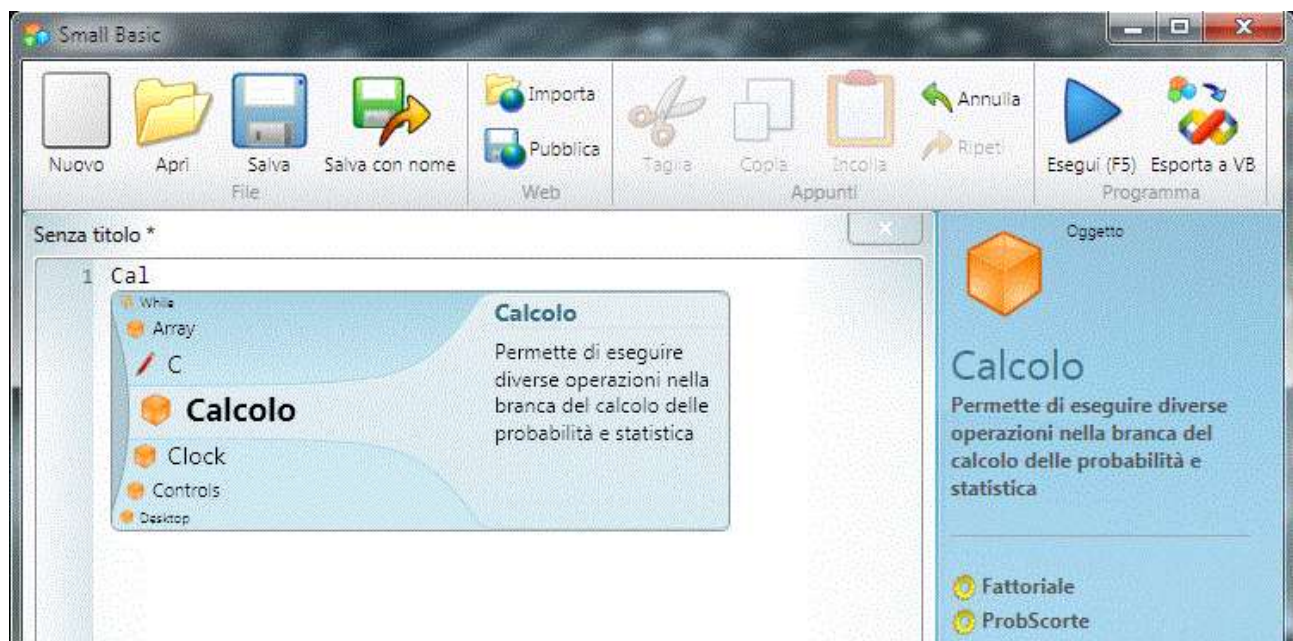
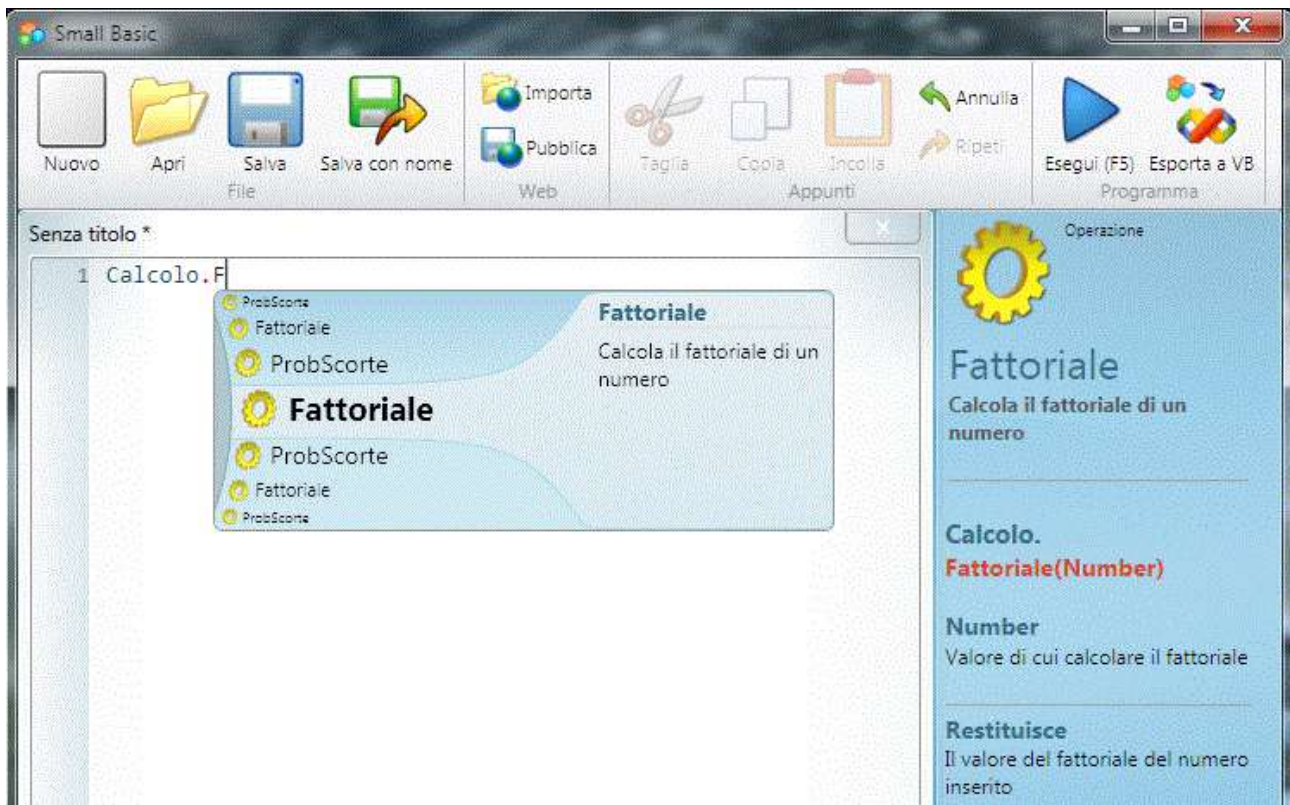
È un metodo che consente di estendere le applicazioni dotandole della possibilità di caricare componenti esterni, utile per fare crescere le applicazioni.
Per plugin s'intende una serie di **DLL** (*Dynamic Link Library*) che una volta caricate nell'applicazione implementano nuove funzionalità.
È utile poichè consente di espandere le funzionalità base dell'applicazione anche a chi non ne è lo sviluppatore iniziale, posto che chi sviluppa il plugin deve condividere con l'applicazione base una serie di regole che consentono di fare dialogare i due elementi.

È un componente software non necessariamente sviluppato dallo stesso team che ha creato l'applicazione che lo ospita, chiamata host.
Non necessariamente scritto nello stesso linguaggio.
È caricato dall'host a run-time tramite **late binding** e utilizzato senza che l'applicazione padre debba essere ricompilata.

Il fatto che un plugin possa essere scritto da un team diverso da quello che ha scritto l'applicazione host, significa che quest'ultima dovrà mettere a disposizione una serie di **API** (*Application Programming Interface*) che altri programmatori possono utilizzare per sviluppare i plugin.
Il .NET Framework garantisce la compatibilità binaria tra assembly sviluppati in linguaggi diversi che comunque generano codice **IL** (*Intermediate Language*).
Il fatto che il plugin sia caricato a run-time dà la possibilità di "agganciarlo" all'applicazione host senza che questa ne sia a conoscenza a priori, in altre parole non esiste un reference diretto all'assembly del plugin durante la compilazione dell'applicazione host.
Per essere espandibile, l'applicazione deve avere qualche meccanismo per ricercare, caricare e attivare i plugin che sono stati sviluppati appositamente per essa.
La prima operazione che l'applicazione host deve compiere è capire dove sono e quali sono i plugin da caricare.
Per risolvere questo problema si ricorre a un file di configurazione **XML** (*eXtensible Markup Language*) che contiene la lista dei file da caricare.
L'applicazione legge il file XML e ottiene la lista dei plugin.
I plugin di Small Basic devono essere memorizzati nella cartella seguente.
C:\PROGRAM FILES (X86)\SMALLBASIC\LIB.

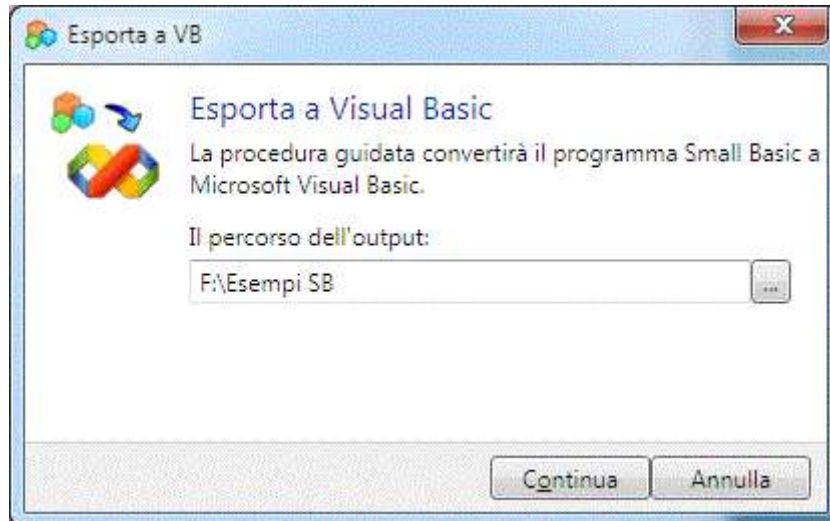
Esempio, plugin per il calcolo del fattoriale.

```
TextWindow.Show()
TextWindow.Title ="Selezione"
TextWindow.Clear()
TextWindow.Write("Inserisci un numero intero: ")
n = TextWindow.ReadNumber()
fat = Calcolo.Fattoriale(n)
TextWindow.WriteLine("")
TextWindow.ForegroundColor="Yellow"
TextWindow.WriteLine("Il fattoriale vale: "+fat)
TextWindow.ForegroundColor="Gray"
TextWindow.CursorLeft =1
TextWindow.CursorTop =23
```

Eseguire l'applicazione.

L'errore visualizzato è dovuto alla mancata memorizzazione della libreria LIBRERIACALCOLO.DLL nella cartella indicata in precedenza.

Chiudere l'ambiente di sviluppo e riaprirlo, verificare che il nuovo oggetto sia presente.

Eseguire l'applicazione.

CONVERSIONE IN VISUAL BASIC

INTRODUZIONE

Aprire un’applicazione e premere il pulsante ***Esporta a VB***.
Per prima cosa bisogna indicare in quale cartella si vuole ottenere il progetto convertito.

Una volta premuto il pulsante ***Continua***, si aprirà automaticamente Visual Studio e partirà il wizard della conversione del progetto.

Conversione guidata di Visual Studio

Conversione guidata di Visual Studio

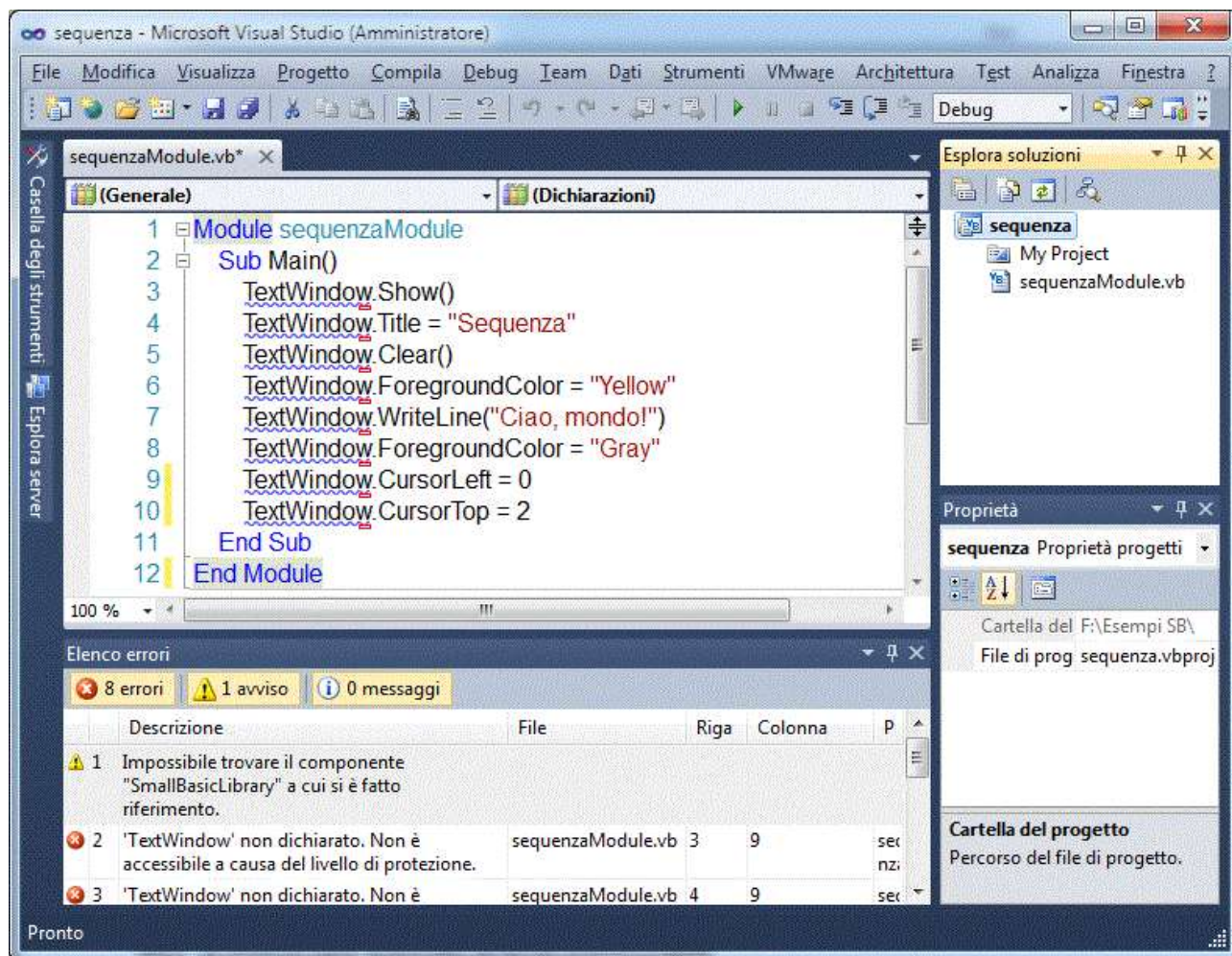
Deve essere eseguita la conversione della soluzione o del progetto che si sta tentando di aprire in quanto è stato creato in una versione precedente di Visual Studio o si basa su una versione meno recente di un programma installato nel computer. In seguito alla conversione di una soluzione o dei relativi progetti, potrebbe non essere più possibile modificare, compilare o eseguire la soluzione nelle versioni precedenti.

Se la soluzione o il progetto è sottoposto al controllo del codice sorgente, ne verrà eseguita automaticamente l'estrazione durante la conversione. Verificare che il plug-in del controllo del codice sorgente corretto sia attivo e che nessun file venga estratto in modo esclusivo da altri utenti.

Se la soluzione o il progetto non è sottoposto al controllo del codice sorgente, verificare che per tutti i file siano disponibili autorizzazioni di lettura/scrittura.

Scegliere Avanti per continuare.

< Indietro | Avanti > | Fine | Annulla

Confermare tutti i passi del wizard, mantenendo le opzioni di default.
Non tutto funzionerà subito a dovere.

L’errore principale è dato dalla mancanza del file di libreria SMALLBASICLIBRARY.DLL. La libreria non è aggiunta automaticamente e quindi dev’essere aggiunta manualmente. Aprire il menu contestuale con un clic destro sul nome del progetto, nella finestra ***Esplora Soluzioni***, poi si dovrà selezionare la voce ***Aggiungi\Elemento esistente… (CTRL+D)***.
Si apre una finestra per la selezione del file da aggiungere, se i file DLL non sono visibili nella finestra, è necessario modificare il filtro dei file da visualizzare.

Una volta confermata la selezione, la DLL è aggiunta al progetto: la libreria è referenziata e gli errori spariscono.

Eseguire l'applicazione.